

**UNIVERSIDAD AUTÓNOMA DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**



## **TRABAJO FIN DE GRADO**

**Herramienta móvil para el guiado de personas con  
discapacidad cognitiva en entornos interiores**

**Javier López García**

**JUNIO 2015**

Tutor: Javier Gómez Escribano

Ponente: Germán Montoro Manrique



## Resumen

A las personas con diversidad funcional intelectual, la orientación supone un gran esfuerzo, dado que tienen muchos problemas para el desplazamiento por lugares desconocidos o para recordar actividades diarias. Integrarse en todos los ámbitos de la sociedad es un reto para estas personas. Desarrollar habilidades sociales, así como conseguir un puesto de trabajo hacen que su adaptación sea difícil.

En relación con estos problemas se ha desarrollado un sistema que pretende guiar a usuarios con diversidad funcional intelectual por entornos interiores. Se ha diseñado una aplicación móvil para ello. El sistema operativo elegido ha sido Android por la gran facilidad que existe hoy en día de hacerse con un dispositivo móvil de estos. La interfaz de usuario ha sido diseñada con el fin de ser fácil de utilizar para estas personas. De esta forma se conseguirá una independencia del usuario sobre el tutor. Además, el colectivo de usuarios al que se dirige la aplicación hace que tenga que ser intuitiva. No debe provocar dudas en el usuario.

El sistema utiliza un *smartphone* por sus características, tanto de software como de hardware. Para la localización de destinos se emplean etiquetas QR. Los motivos por los que se ha elegido esta tecnología son su bajo coste y su gran facilidad para ser implantado en diferentes entornos.

Para esta aplicación se ha seguido un diseño iterativo, incremental y asistido por expertos, tanto en el diseño de interfaces como en la educación. Este planteamiento se asemeja al DCU (diseño centrado en el usuario), pero dadas las características del usuario se ha consultado con expertos. Esto es así puesto que en este tipo de sistemas la evaluación por parte del cliente es muy importante. La realimentación se basará en los educadores y su experiencia con los usuarios. Es decir, se atiende a las necesidades de estos para el desarrollo del sistema.

Como principal línea de trabajo futuro queda esta evaluación con usuarios con diversidad funcional intelectual para obtener resultados fieles a la realidad. Las conclusiones obtenidas de esta evaluación, nos proporcionarán información suficiente para conseguir una aplicación mejor adaptada a las necesidades de los usuarios.

## Palabras Clave

Android, dispositivo móvil, diversidad funcional, código QR, Smartphone, entornos interiores



## **Abstract**

For people with cognitive disability, guidance is a big effort, given that they have so many troubles moving across unknown places, or remembering daily activities. To integrate in all areas of society is such a challenge for them. To develop social skills, as far as getting a job, are things that make their adjustment harder.

Related to these problems, a system has been developed which tries to guide cognitive disabled people across indoor areas. A mobile application has been designed for this. The chosen OS has been Android, because it is easy nowadays to acquire that kind of device. The UI has been designed with the aim of being easy to use for these people. In this way, the user will be more independent from his guardian. Additionally, the group of users the application is directed to, needs it to be intuitive. It shouldn't make them doubt.

The system uses a smartphone, because of its features, as much as from software as from hardware. To find the destinations it uses QR tags. The reasons for that choice are the low prices and because they are easy to adapt to different environments.

The design is focused on the user. This is so because in these types of systems customer evaluation is very important. The feedback is based on educators and experience with users. So, the development of the system is focused on the users needs.

Still remains to be done this evaluation in the future, with these kinds of users, to get results according to real life. The evaluation's conclusions, will provide us enough information to achieve an application better adapted better application in the future, to the user's needs.

## **Keywords**

Android, mobile device, cognitive disability, QR code, smartphone, indoor areas



# Índice de Contenidos

<b>1 INTRODUCCIÓN .....</b>	<b>1</b>
1.1 MARCO DEL PROYECTO.....	1
1.2 MOTIVACIÓN.....	1
1.3 OBJETIVOS DEL PROYECTO.....	2
1.4 ESTRUCTURA DEL DOCUMENTO.....	3
<b>2 DIVERSIDAD FUNCIONAL INTELECTUAL.....</b>	<b>5</b>
2.1 ¿QUÉ ES? .....	5
2.2 CAUSAS DE LA DIVERSIDAD FUNCIONAL INTELECTUAL.....	6
2.3 GRADOS DIVERSIDAD FUNCIONAL INTELECTUAL.....	6
2.4 APORTACIÓN DEL PROYECTO .....	7
<b>3 ESTADO DEL ARTE .....</b>	<b>9</b>
3.1 INTRODUCCIÓN .....	9
3.2 CONCLUSIONES .....	11
<b>4 DEFINICIÓN DEL PROYECTO .....</b>	<b>13</b>
4.1 METODOLOGÍA .....	13
4.2 MODELO DE CICLO DE VIDA .....	13
4.3 MODELO DE DESARROLLO .....	14
4.4 HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS .....	14
<b>5 ANÁLISIS Y DISEÑO .....</b>	<b>17</b>
5.1 INTRODUCCIÓN .....	17
5.2 ORÍGENES DE LA APLICACIÓN .....	18
5.3 ANÁLISIS DE REQUISITOS .....	21
5.4 DISEÑO .....	22
<b>6 IMPLEMENTACIÓN .....</b>	<b>27</b>
6.1 ARQUITECTURA DEL SISTEMA .....	27
6.2 SERVIDOR .....	27
6.3 CLIENTE .....	28
6.4 BASE DE DATOS.....	41
<b>7 PRUEBAS .....</b>	<b>45</b>
7.1 PRUEBAS CLIENTE .....	45
7.2 PRUEBAS SERVIDOR .....	48
<b>8 CONCLUSIONES.....</b>	<b>51</b>
8.1 TRABAJO FUTURO.....	52

<b>9 BIBLIOGRAFÍA.....</b>	<b>53</b>
<b>ANEXO A: ALGORITMO BREADTH FIRST SEARCH .....</b>	<b>A</b>
<b>ANEXO B: VOLCADO BASE DE DATOS.....</b>	<b>C</b>



## Índice de Ilustraciones

ILUSTRACIÓN 1: MODELO DE CICLO DE VIDA .....	14
ILUSTRACIÓN 2: DISEÑO ITERATIVO.....	18
ILUSTRACIÓN 3: UBICACIÓN LOCAL EN QRUMBS.....	19
ILUSTRACIÓN 4: IMÁGENES QRUMBS.....	19
ILUSTRACIÓN 5: PANTALLA DE CARGA DE ASSIST-IN.....	23
ILUSTRACIÓN 6: LISTA DE USUARIOS Y BÚSQUEDA DE CÓDIGO QR .....	23
ILUSTRACIÓN 7: LISTA DE DESTINOS DISPONIBLES.....	24
ILUSTRACIÓN 8: BÚSQUEDA DE DESTINO .....	24
ILUSTRACIÓN 9: PISTA CORRECTA.....	25
ILUSTRACIÓN 10: PISTAS INCORRECTAS.....	25
ILUSTRACIÓN 11: DESTINO CORRECTO.....	26
ILUSTRACIÓN 12: ARQUITECTURA CLIENTE-SERVIDOR.....	27
ILUSTRACIÓN 13: IMÁGENES FINALES DE LA APLICACIÓN.....	30
ILUSTRACIÓN 14: ESTRUCTURA CLIENTE ASSIST-IN.....	31
ILUSTRACIÓN 15: CONSTRUCTOR ROUTEINFOPROVIDER.....	32
ILUSTRACIÓN 16: DESCARGA DE IMÁGENES USERIMAGESDOWNLOADER .....	33
ILUSTRACIÓN 17: CONEXIÓN CON SERVIDOR .....	33
ILUSTRACIÓN 18: TIPO DE INFORMACIÓN REGISTRO DE ACTIVIDAD.....	34
ILUSTRACIÓN 19: MÉTODO GETCONTENTS .....	36

ILUSTRACIÓN 20: MÉTODO CHECKCONEX .....	37
ILUSTRACIÓN 21: MÉTODO INFLATEDESTINATIONLIST .....	37
ILUSTRACIÓN 22: LLAMADA A MÉTODO GENERAGRAFO .....	38
ILUSTRACIÓN 23: GRAFO REPRESENTATIVO DE MAPA DE ASSIST-IN .....	38
ILUSTRACIÓN 24: MÉTODO ONINIT DE TEXTTOSPEECH.....	40
ILUSTRACIÓN 25: MÉTODO ONDESTROY DE TEXTTOSPEECH .....	41
ILUSTRACIÓN 26: ESQUEMA RELACIONAL DE BASE DE DATOS.....	41
ILUSTRACIÓN 27: FICHERO DUMP.PHP .....	49
ILUSTRACIÓN 28: VOLCADO DE LA BASE DE DATOS .....	C

## Índice de Tablas

TABLA 1: TABLA DE NODOS DE LA BASE DE DATOS .....	42
TABLA 2: TABLA DE MAPAS DE LA BASE DE DATOS .....	42
TABLA 3: TABLA DE ARISTAS ENTRE NODOS .....	43
TABLA 4: MATRIZ DE UN GRAFO REPRESENTATIVO DE UN MAPA.....	B

## Glosario

- **Smartphone:** Palabra de origen anglosajón. Teléfono móvil inteligente construido sobre una plataforma informática móvil. Con gran capacidad de almacenamiento de datos, similar a una minicomputadora.
- **Android:** Sistema operativo diseñado para teléfonos móviles basado en el *kernel* de Linux.
- **Kernel:** Es el núcleo de un sistema operativo.
- **Arquitectura software:** Indican la estructura, funcionalidad e interacción entre las diferentes partes que forman el software.
- **Cliente:** Aquella aplicación o computadora que necesita un servicio en otro computador.
- **Servidor:** Aplicación en ejecución que atiende las necesidades (peticiones) de un cliente.
- **Script:** Conjunto de instrucciones almacenadas por lo general en texto plano, interpretados secuencialmente para su ejecución.
- **Diversidad funcional intelectual:** Término alternativo al de discapacidad con el que se pretende no generar exclusiones o discriminaciones.
- **Breadth First Search:** Algoritmo de búsqueda en anchura. Recorre los elementos de un grafo desde la raíz en búsqueda de sus nodos adyacentes hasta el final del grafo.
- **Código QR:** Palabra anglosajona cuyo significado es “quick response code”, código de respuesta rápida. Almacena información en una matriz de puntos o código de barras bidimensional.

# 1 Introducción

---

## 1.1 Marco del proyecto

El Trabajo de Fin de Grado descrito en este documento tiene como propósito implementar una herramienta móvil para el guiado de personas con discapacidad cognitiva en entornos interiores. Para ello se utiliza etiquetado de localizaciones clave basado en códigos QR. Cada punto del entorno se marca mediante dichas etiquetas. Para informar al usuario de cómo llegar al siguiente punto de la ruta, se muestra una imagen indicando el siguiente destino. Existe un visionado directo entre los dos puntos para una más fácil comprensión.

El mapa de la aplicación se modela mediante un grafo. Cada nodo corresponde con un punto del entorno. Los destinos pueden tratarse de localizaciones intermedias o finales, consideradas como destinos a los que el usuario se puede dirigir. Para la ruta se implementa el algoritmo BFS, que calcula el camino a seguir por el usuario.

## 1.2 Motivación

Desde la Organización Mundial de la Salud (O.M.S.) [1], se hace referencia a las personas con diversidad funcional como “personas con discapacidad”. El propio colectivo ha iniciado una corriente en [2], con la que se pretende eliminar los términos negativos y que generan exclusión o discriminación. El objetivo es no crear barreras a estas personas, permitiendo que la integración en la sociedad sea más sencilla.

Las nuevas tecnologías ayudan en gran medida a esta integración. En este proyecto nos centramos en las tecnologías de asistencia, encargadas de hacer más fácil la realización de tareas cotidianas para estas personas. Concretamente, y como se explicará más adelante, este Trabajo de Fin de Grado (TFG) se centra en las tecnologías de asistencia clasificadas como de **alta tecnología**, basadas en diferentes ingenierías y en las TIC (Tecnologías de la Información y la Comunicación). Se pretende utilizar estas tecnologías (un teléfono inteligente, *smartphone*) como herramienta cotidiana para que el usuario vaya adquiriendo independencia en la tarea de desplazarse en un entorno de interiores.

En la actualidad se trabaja para que personas con diversidad funcional consigan integrarse en la sociedad en todos sus ámbitos. Estos van desde el trabajo, fundamental para el desarrollo de la persona, hasta la educación, comunicación, ocio, etc.

En el laboratorio de inteligencia ambiental (*AmILab*) [3] de la Universidad Autónoma de Madrid (U.A.M), se ha desarrollado el proyecto AssisT, que ofrece asistencia integral a

personas con diversidad funcional intelectual. La aplicación móvil *QRumbs* [4] forma parte de este proyecto, con el objetivo de guiar a estas personas en espacios interiores. Se trata de una aplicación móvil obsoleta y con una escasa funcionalidad.

El proyecto surge de la necesidad de desarrollar una aplicación completa para guiar a personas con diversidad funcional intelectual en entornos interiores. Se realizará un nuevo enfoque, desde la funcionalidad a implementar hasta un nuevo diseño de la interfaz de usuario.

### **1.3 Objetivos del proyecto**

Este proyecto se va a desarrollar con el objetivo principal de asistir a las personas con diversidad funcional intelectual en sus actividades diarias, en este caso, en sus desplazamientos en entornos interiores. En adelante en esta memoria, nos referiremos a este tipo de entornos. Por tanto, la aplicación deberá cumplir con los siguientes objetivos:

- **Sencilla de usar.** Es un objetivo fundamental para las personas con diversidad funcional intelectual. El aprendizaje de la aplicación debe ser rápido.
- **La interfaz de usuario** y su funcionalidad deben adaptarse a estas personas. Esto es, debe ser lo más intuitiva posible.
- El usuario de la aplicación debe ir ganando **confianza e independencia** con la experiencia de uso. Se busca que la aplicación sirva para ser lo más independientes posibles en la búsqueda del objetivo. Según el perfil del usuario y sus características variará el tiempo de adaptación a la aplicación.
- Otro objetivo importante de la aplicación será la **robustez**. El sistema no debe fallar durante su uso. Esto ayudará al usuario la siguiente vez que utilice la aplicación dado que le irá dotando de más confianza en sí mismo para cumplir con su objetivo.
- El **coste** es otro objetivo importante en cualquier proyecto. Puesto que se usarán códigos QR<sup>1</sup> [5] para las localizaciones, se prevé que el coste de implantación en el entorno no sea elevado.
- En cuanto al **mantenimiento** de la aplicación, el objetivo irá en función de la experiencia de uso de los usuarios. Debe adaptarse a las características y cualidades del colectivo al que se dirige, por lo que se estima que no sea necesario un mantenimiento exhaustivo. Se necesitarán de los datos y

---

<sup>1</sup> Denso Wave Incorporated, "code QR," Software, 2007

resultados que se obtengan de los usuarios para perfeccionar determinados detalles de la aplicación.

- Es importante también que la ruta propuesta por el sistema para llegar al destino elegido sea **fiable**. El usuario se desentiende y se fía de la aplicación para poder llegar a su destino. Es muy importante que esta ruta no dé lugar a errores o lleve al usuario a una confusión.

## **1.4 Estructura del documento**

En este apartado del capítulo se resume el contenido de esta memoria. A continuación se listan los capítulos que la componen:

- El **capítulo 2** trata el tema central del proyecto. Aquí se explica de manera resumida qué es la diversidad funcional intelectual, sus causas principales y los diferentes niveles existentes.
- En el **capítulo 3**, se hace una revisión del estado del arte. Se explican las tecnologías de asistencia, centrándonos en las principales que usaremos en este proyecto.
- El **capítulo 4** es la definición del proyecto. Aquí se explica la metodología, ciclo de vida y modelo de desarrollo seguidos. Además se detallan las tecnologías y herramientas utilizadas para el desarrollo del proyecto.
- En el **capítulo 5** se hace una descripción del sistema. Se explica la arquitectura utilizada y el diseño de la aplicación.
- El **capítulo 6** explica cómo se ha implementado el proyecto. Se detalla la estructura de cliente y servidor, así como los módulos que componen la aplicación.
- En el **capítulo 7** se explican las pruebas realizadas en el proyecto.
- El **capítulo 8** es un conjunto de conclusiones sobre el proyecto realizado. Además, se explica el trabajo futuro a realizar.





# 2 Diversidad Funcional Intelectual

---

## 2.1 ¿Qué es?

Ya se ha comentado la corriente existente para eliminar los términos negativos de textos, artículos, estudios, etc., para referirnos a personas con diversidad funcional intelectual. Según la A.A.I.D.D. (*American Association on Intellectual and Developmental Disabilities*) [6], la discapacidad intelectual se definía como:

*“Es una discapacidad caracterizada por limitaciones significativas en el funcionamiento intelectual y en la conducta adaptativa que se manifiesta en habilidades adaptativas conceptuales, sociales, y prácticas”*

Las personas con diversidad funcional intelectual tienen una limitación en el desarrollo de estas habilidades, lo que hace que tengan más dificultades para afrontar diferentes situaciones que otras personas. Esto puede variar desde el tiempo de aprendizaje para determinadas tareas, hasta la imposibilidad total de realizarlas, en función de las características del individuo.

La A.A.I.D.D. hace hincapié en que estas limitaciones han de ser bien medidas y valoradas. Deben considerarse en un contexto habitual para personas de la misma edad. También, se debe tener en cuenta la diversidad cultural para una evaluación correcta. Y, por último, es importante desarrollar unas técnicas de apoyo. Las personas con diversidad funcional, experimentarán mayores progresos si disponen de un apoyo en tareas que suponen cierto esfuerzo. Esto ayudará a que las mismas actividades se puedan ir realizando cada vez con menor dificultad.

El nivel y funcionamiento intelectual de una persona se relaciona con las diferentes habilidades intelectuales del individuo, con la conducta social y conceptual de la persona, la salud física y mental, así como con el entorno y la cultura de dicho entorno. Estas forman las diferentes dimensiones de la discapacidad intelectual:

- Habilidades intelectuales.
- Conducta adaptativa. Conceptual, social y práctica.
- Participación, interacciones y rol social.
- Salud física y mental.
- Contexto. Ambiente y cultura.

## **2.2 Causas de la diversidad funcional intelectual**

Los factores que afectan a la diversidad funcional pueden ser diferentes y pueden darse de forma conjunta. De forma individual, todos pueden ser causa de una diversidad funcional en mayor o menor grado. Estos factores son:

1. **Factores biomédicos.** Dentro de ellos, se abarcan muchos factores como la alteración de cromosomas, anomalías genéticas y trastornos metabólicos, algunas enfermedades infecciosas o traumatismos craneoencefálicos. Entre las alteraciones cromosómicas encontramos trastornos genéticos conocidos como el síndrome de *down* o el síndrome del maullido de gato. Los trastornos metabólicos son alteraciones al procesar sustancias necesarias para el organismo, por ejemplo, el síndrome de *Hurler*.
2. **Factores sociales.** Los más habituales son la malnutrición materna, la pobreza y la falta de una buena educación por parte de los progenitores.
3. **Factores de conducta.** Estos incluyen el consumo de drogas y alcohol por parte de los padres, violencia doméstica o maltrato hacia el hijo.
4. **Factores educativos.** El principal es la falta de apoyo del entorno, familiar y social.

## **2.3 Grados diversidad funcional intelectual**

Puesto que se tiene como objetivo dotar de una mayor autonomía e independencia al usuario, es de vital importancia un análisis de los grupos existentes. Según la O.M.S. se clasifican en:

- **Leve** (CI: 70-85): En este nivel, las personas con estos niveles de coeficiente intelectual sufren un retraso en el tiempo de aprendizaje, pero no se verían limitados para el uso de la aplicación.
- **Moderada** (CI: 50-70): Es el nivel ideal que debe tener un usuario al que se destina la aplicación. Las personas que se encuentran en este grupo de coeficiente intelectual, desarrollan habilidades sociales y comunicativas, así como, una cierta dificultad en el entorno laboral. No desarrollan problemas graves de atención y adaptación al entorno.
- **Grave** (CI: 25-40): Este sería el nivel mínimo de coeficiente intelectual que debe tener un usuario de la aplicación. Son personas capaces de ser autónomos pero con una gran dificultad. Por este motivo, seguramente, debería recibir un apoyo y seguimiento durante el uso de la aplicación.

- **Profunda** (CI: 20- 25): Personas con este nivel de coeficiente intelectual presentan graves problemas sensoriales y motrices. Estas personas, no serían capaces de adaptarse al entorno en el que se implantaría el proyecto.

La inserción en la sociedad de este colectivo es uno de los principales objetivos que ponen en marcha en España fundaciones como ONCE (Organización Nacional de Ciegos Españoles). Se fomenta el empleo para que las personas se integren en el ámbito del empleo así como la necesidad de que estas personas se sientan útiles para la sociedad.

Uno de los mayores obstáculos para estas personas es la falta de información sobre la diversidad funcional intelectual. En el Real decreto legislativo 1/2013 [7], artículos 42 y siguientes, se regula el derecho al empleo de personas con diversidad funcional intelectual. En el artículo 42, se establece que las empresas con 50 o más trabajadores, deben tener al menos un 2% de su plantilla formada por personas de este colectivo. Además, existen diferentes artículos cuyo objetivo es favorecer la integración de estas personas al ámbito laboral. Por ejemplo, el artículo 44 menciona como los centros especiales de empleo pueden verse beneficiados por compensaciones económicas en favor de la viabilidad de estos centros.

En la mayoría de empresas el proceso que llevan a cabo es lento y complicado. Suelen recibir formación específica para realizar las tareas que les han sido encomendadas. Es muy común que estas personas reciban cursos previos de formación para facilitar la adaptación.

## ***2.4 Aportación del proyecto***

Las personas con diversidad funcional intelectual, en función de las características de cada una, se enfrentan a diario con dificultades a la hora de realizar tareas cotidianas. Entre estas actividades se encuentra la tarea de llegar a un destino en un entorno de interiores.

La intención de este proyecto es que el usuario logre su objetivo por sí mismo, ganando confianza para conseguir un grado de independencia respecto a su tutor. Esta le ayudará en sus próximas tentativas con la aplicación. El aprendizaje será progresivo, de tal forma que la implicación del educador vaya disminuyendo con el paso del tiempo.

En el ámbito laboral, estas personas tienen dificultades también para la inserción. Se pretende que el usuario confíe más en sí mismo, y por consiguiente, vaya adquiriendo unas determinadas capacidades que le ayuden en su integración.

El acceso a un teléfono inteligente hoy en día es muy sencillo. La disponibilidad de terminales que cumplen los requisitos de esta aplicación son enormes. Se considera importante el uso de un *smartphone* como herramienta principal dada la cantidad de tareas que nos facilita. Esto ayudará al usuario en su familiarización con las nuevas tecnologías.



# 3 Estado del arte

---

## 3.1 Introducción

A diario vemos como los avances tecnológicos van llegando a diferentes sectores como la educación, la medicina, el transporte, etc. De igual forma, la tecnología ha evolucionado para facilitar la vida de personas con diversidad funcional.

Las tecnologías de asistencia (AT, *Assistive Technology*) vienen definidas en [8]:

*“Cualquier artículo, equipo global o parcial, e incluso, cualquier sistema adquirido comercialmente o adaptado a una persona, el cual se usa para mejorar y/o incrementar las capacidades funcionales de individuos con discapacidades, así como de modificar o instaurar conductas”*

Estas tecnologías son tan amplias que dividen en 4 categorías o grupos diferentes en función del nivel de la tecnología utilizada:

- **Sin tecnología:** En este grupo se incluyen, por ejemplo, tamaños de letras más grandes o técnicas de estudio.
- **Baja tecnología:** Utilizan adaptaciones de herramientas ya existentes.
- **Media tecnología:** Aquí se incluyen, por ejemplo, sillas de ruedas. Son equipos y elementos de cierta dificultad tecnológica.
- **Alta tecnología:** Son equipos de tecnología avanzadas de gran complejidad. Basados, principalmente, en las TIC (Tecnologías de la información y comunicación) y diferentes ingenierías. Sirva como ejemplo de estos elementos, las sillas de ruedas autónomas.

En este proyecto nos vamos a centrar en las tecnologías de asistencia de alta tecnología. Más concretamente, en las que se relacionan con los dispositivos móviles.

En este ámbito, la fundación Auna realizó un estudio [9] en 2003 sobre el impacto de las tecnologías en personas con diversidad funcional intelectual. También se estudió la actitud con la que estas personas se enfrentaban a las nuevas tecnologías, cuyos resultados son muy positivos. Se extrae de los resultados obtenidos que las tecnologías con mayor impacto son, en este orden:

1. Ordenador
2. Teléfono y dispositivos móviles
3. Internet

En este proyecto nos centramos en la asistencia mediante los teléfonos inteligentes de hoy en día, en su mayoría con conexión a internet.

David J. Browne *et al.* [10] investigan con juegos de aprendizaje así como sistemas de localización para ayudar a personas con diversidad funcional intelectual. Se sigue un desarrollo centrado en el usuario, puesto que la realimentación por parte de éste hará que el sistema final se acerque más a sus necesidades.

Más análisis realiza Steffan Carmien *et al.* [11]. Este estudio tiene como idea principal que los usuarios con diversidad funcional intelectual sean más independientes, y propone que los educadores vayan teniendo menos importancia en su evolución. Su objetivo es desarrollar proyectos software que doten de independencia al usuario en la medida de lo posible. La función de los educadores debe ser solo colaborativa en la implementación del proyecto.

D.Braddock *et al.* [12] trata de separar las tecnologías de asistencia en dos grupos. Por un lado las que tienen por objetivo el apoyo personal, y por otro tecnologías como robots o de realidad virtual.

A continuación nos centramos en los proyectos desarrollados sobre la diversidad funcional que ayudan al usuario a guiarse por un entorno hasta llegar a su objetivo.

El proyecto “ARGUS” [13] se ha desarrollado para guiar a personas con diversidad funcional visual a través de la tecnología. En este software, se guía al usuario a través de sonidos binaurales, que tras diversos estudios se ha visto que pueden ayudar a estas personas a percibir la dirección correcta así como la distancia.

Otro proyecto español que encontramos es “Ver con las manos” [14] de la Universidad Rey Juan Carlos. Este proyecto nació para guiar a personas con diversidad funcional tanto en entornos interiores como exteriores, pero posteriormente vieron que podría ser de utilidad para guiar a turistas en ciudades extranjeras. En este software, la gestión de rutas se hace a través de una página web, y posteriormente, el usuario puede utilizar estas rutas para guiarse. Son las entidades e instituciones las encargadas de crear las rutas para los usuarios en función de las necesidades de cada uno. Este proyecto no ha sido probado con personas con un tipo de diversidad funcional intelectual sino únicamente visuales y auditivos. La principal ventaja de este proyecto es que no necesita de señal GPS para el guiado de las personas.

Para personas con diversidad funcional intelectual se han desarrollado proyectos que tienen como objetivo acercar las nuevas tecnologías a este colectivo. Entre estos se encuentra el “Proyecto CITI (Competencia Intelectual y Tecnología de la Información)” de la fundación Orange [15]. Se trata de un juego para *smartphone*, que pretende además de que el usuario se familiarice con la tecnología y mejore sus habilidades y capacidades cognitivas, también disfrute de un tiempo de ocio y diversión. Otro proyecto en este sentido es el “ARCA de los pensamientos” [16]. Aquí se pretende desarrollar y entrenar

las habilidades más importantes para pensar y aprender, como son la memoria, atención o el razonamiento. Estos dos proyectos forman parte de un proyecto mayor que recibe el nombre de proyecto BIT (Bases Informáticas Tecnológicas). Consiste en una serie de proyectos dirigidos a este tipo de usuario, con el objetivo de que se puedan integrar con más facilidad social, educativa y laboralmente.

En cuanto al estudio del guiado de estas personas por diferentes entornos, encontramos proyectos como “WADER” [17] . Se trata de una aplicación con características comunes al proyecto propuesto. Entre ellas el sistema de guiado se realiza a través de códigos QR, y se utiliza una PDA como herramienta principal.

Otro proyecto cuyo objetivo es guiar a este colectivo de personas utilizando el transporte público de una ciudad es [18]. También se utiliza una PDA como herramienta principal para el usuario. Requiere de conexión a internet y señal GPS para proporcionar las indicaciones. En este sistema se utilizan indicaciones visuales así como auditivas para un mejor guiado. Los resultados obtenidos son satisfactorios, donde un 73% de los usuarios que probó el sistema consiguió salir del autobús en la parada correcta.

### **3.2 Conclusiones**

En este apartado de la memoria se han analizado estudios y diferentes proyectos genéricos para personas con diferentes tipos de diversidad funcional, incluida intelectual.

Los proyectos a desarrollar para este colectivo de personas han de ser centrados en el usuario. De esta forma las conclusiones obtenidas serán más fiables. Las pruebas que se realicen durante el desarrollo del proyecto, deben realizarlas aquellos a los que se dirija el proyecto, dado que así los desarrolladores y encargados del proyecto podrán acercarse más a las necesidades de cada usuario.

De diferentes estudios se sabe que las personas con diversidad funcional intelectual en su mayoría tienen habilidades y capacidades suficientes para el uso de las nuevas tecnologías como el *smartphone*. Para aquellos con un grado mayor de afectación, será necesario la colaboración de los educadores y/o tutores. En este proyecto, para ayudar a su independencia, la importancia del tutor irá disminuyendo en función del aprendizaje del usuario, hasta que su importancia sea mínima. El uso de un *smartphone* como herramienta hace más fácil la implantación del sistema en diferentes entornos. En otros proyectos similares se utilizaban herramientas como una PDA. Las características y recursos, como la memoria o cámara de fotos, además de la facilidad de hacerse con uno hoy en día favorecen esta implantación.

En cuanto a la interfaz, el objetivo del proyecto será hacerla lo más intuitiva posible, como se hace en proyectos como “CITI”. Dado que se sabe que para determinado tipo de usuarios se necesita una interfaz sencilla, se trata de un requisito imprescindible atendiendo al colectivo al que se dirige la aplicación.

Las indicaciones que se irán dando al usuario han de ser de fácil comprensión. No deben confundirle a la hora de tomar decisiones. Otros proyectos, como “WADER”, utilizan indicaciones con flechas, lo que se ha demostrado que confunde en determinadas ocasiones, puesto que puede requerir de cierta interpretación. En nuestro caso se van mostrando las indicaciones por cada punto que debe ser visitado, restando complejidad.

La tecnología utilizada para estas indicaciones es una decisión importante. En este proyecto se utilizan las etiquetas QR para indicar los puntos clave de una ruta. En otros proyectos analizados se usa, por ejemplo, los sonidos binaurales. Aunque se ha demostrado que para el guiado de personas es viable, para este TFG se adapta mejor las etiquetas QR, por su bajo coste y su facilidad de implantación. Otros proyectos utilizan y necesitan para su funcionamiento de señal GPS, algo que en espacios interiores como en los que se pretende implantar este proyecto, puede no llegar a funcionar correctamente, ya que es común la pérdida de señal en estos entornos.

Para facilitar el guiado del usuario se ha añadido un asistente de voz. Además de las pistas visuales que debe seguir, se le irá guiando mediante pistas de audio que facilitan la comprensión en determinados momentos. Este detalle no se realiza en proyectos como “WADER”. En este proyecto tampoco se guarda un histórico de eventos del usuario de la aplicación. Esto ayudará a los tutores a desarrollar mejor las rutas con el fin de facilitar su trayecto.

Otro punto importante es la conexión a internet. En muchos proyectos se requiere de forma imprescindible. En nuestra aplicación se ha pensado que, puesto que en los entornos interiores donde se pretende implantar el sistema puede no haber o perderse la señal con facilidad, una buena opción es no requerir la conexión. En caso de no haber, la aplicación utilizará la última versión descargada en el *smartphone*.

Un detalle que incorpora este proyecto es la personalización de rutas para determinados usuarios. Según los criterios de los educadores y tutores correspondientes, un usuario puede tener una ruta estándar para llegar hasta su destino elegido, o tener configurado ciertos pasos específicos. Esto facilitará su recorrido, puesto que el tutor tiene conocimiento de sus habilidades y puede determinar qué pasos serían más sencillos realizar de diferentes maneras. En alguno de los proyectos analizados en este capítulo, la gestión de rutas se realiza mediante páginas web, sin prestar atención específica a cada usuario



# 4 Definición del Proyecto

---

## 4.1 Metodología

En este apartado del capítulo de definición del proyecto se explica la metodología utilizada durante este proyecto. Se detalla el proceso seguido desde el comienzo del proyecto software. En [19] se define ingeniería del software como:

*“La aproximación sistemática al desarrollo, operación y mantenimiento del software”*

*“Software: Programas de ordenador, procedimientos, reglas, documentación y datos asociados a un sistema de ordenador”*

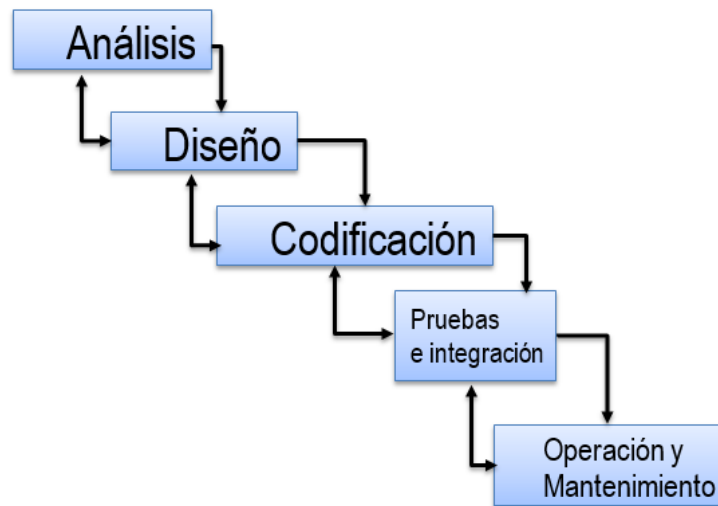
La metodología es un conjunto de métodos que se utilizan en una determinada actividad con el objetivo de formalizarla y optimizarla. Determina qué pasos hay que seguir y cómo han de ser realizados para finalizar una tarea. Centrándonos en un proyecto software como este, en la metodología se define qué hacer, cómo y cuándo ha de hacerse durante todo el proceso de desarrollo y mantenimiento del proyecto.

## 4.2 Modelo de ciclo de vida

Como cualquier proyecto software, este proyecto tiene un ciclo de vida en el que se determina el orden de las fases del proceso, los criterios que definen una transición entre dos fases, y la definición de las entradas y salidas de cada fase.

En este proyecto se ha seguido un modelo de **ciclo de vida de mejora iterativa**. Las características más importantes de este modelo son:

1. Desde cada fase del proceso se puede volver a la anterior en caso de ser necesario.
2. Se puede disponer de diferentes versiones del proyecto sin necesidad de esperar a que todas las fases hayan terminado.
3. Los productos intermedios de cada etapa, se van mejorando.



**Ilustración 1: Modelo de ciclo de vida**

### **4.3 Modelo de desarrollo**

Un modelo de desarrollo de un producto software ayuda a comprender lo que un usuario especifica en los requisitos. Dado que el cliente del producto, por lo general, no sabe especificar completamente sus necesidades, existen modelos para ayudar al desarrollador. Estos modelos se basan en el desarrollo de maquetas o prototipos.

En este proyecto se han desarrollado maquetas para aproximarnos al producto final. Estas maquetas se explican en el apartado 5.4.1.

Las maquetas nos permiten adaptar lo que el usuario quiere, verificar sus requisitos y comprobar la aceptación del producto por parte del usuario.

### **4.4 Herramientas y tecnologías utilizadas**

En este apartado del capítulo de definición del proyecto se explican brevemente las diferentes herramientas y tecnologías utilizadas en este proyecto.

#### **Eclipse**

Eclipse [20] es un entorno de desarrollo que utiliza herramientas de programación de código abierto. Estas herramientas son multiplataforma, soportadas en sistemas operativos como Windows, Linux o Mac.

Es un software complementado por diferentes *plugins* para la programación en diferentes lenguajes. Ejemplos de *plugins* de los más utilizados son el JDT (*Java Development Tools*), o el ADT (*Android Development Tools*) utilizado para el desarrollo de aplicaciones Android como es el caso de este proyecto.

Además se permite añadir a eclipse el *Bundle ADT*, para la creación y desarrollo de aplicaciones Android más rápidamente.

## **Balsamiq Mockups**

Balsamiq Mockups [21] es una herramienta que nos permite la creación de maquetas para interfaces gráficas de usuario.

Este software permite incorporar múltiples plantillas y modelos a los que vienen de serie con la aplicación, para poder generar maquetas con multitud de elementos en dispositivos diferentes (*smartphones* Android, iOS, *tablets*, etc.), con dimensiones de pantalla diferentes.

## **Android SDK**

El Android SDK (*Android Software Development Kit*) [22] se añade al entorno de desarrollo Eclipse para la programación de aplicaciones Android.

Este Kit proporciona al desarrollador herramientas gratuitas como un emulador de Android para realizar las pruebas de una aplicación, un depurador, o el *Manager*. Esta es una herramienta que permite al programador Android la descarga de cualquier versión de librerías Android y su documentación.

## **GenyMotion**

GenyMotion [23] es un emulador de Android disponible para sistemas operativos como Windows, Linux o Mac. En comparación con el emulador nativo de Android, el rendimiento es mayor, lo que facilita la tarea del programador en el desarrollo de una aplicación. Para este mayor rendimiento utiliza la arquitectura x86.

Permite la simulación de diferentes dispositivos móviles (*smartphones* y *tablets*) con diferentes versiones del sistema operativo Android.

## **SequelPro**

Es un gestor de base de datos disponible para los ordenadores Mac. [24] se trata de un gestor con una interfaz muy simple e intuitiva para los desarrolladores, que facilita todas las tareas de gestión de una base de datos.

Tareas tan comunes como actualizar una fila de una tabla, eliminar un campo de una tabla, o insertar una nueva tabla en la base de datos son tareas muy sencillas de realizar mediante la interfaz gráfica de SequelPro. Permite también la exportación e importación de base de datos, lo que permite generar copias de seguridad de nuestra base de datos con gran facilidad.

Se ha utilizado este software para el desarrollo de la aplicación Android de manera local.

## **PHPMyAdmin**

PHPMyAdmin [25] es una herramienta de gestión de bases de datos. Está escrita en lenguaje PHP, y su uso se realiza a través de la web. Permite todas las tareas de gestión de una base de datos como pueden ser crear, eliminar y modificar tablas y sus campos.

En este proyecto se ha utilizado esta herramienta para la gestión de la base de datos de los diferentes proyectos desarrollados en el laboratorio AmILab.

## **OmniGraffle**

OmniGraffle [26] es una herramienta creada para el sistema operativo OS X utilizado en ordenadores Mac.

Es una herramienta que permite la creación de diagramas y gráficas. En este proyecto se ha utilizado esta aplicación para la generación de diferentes diagramas incluidos en esta memoria.

# 5 Análisis y Diseño

---

## 5.1 Introducción

Hoy en día la tecnología avanza sin parar. En sectores como la educación, la medicina o el transporte los avances tecnológicos son notables. En este proyecto nos centramos en una de las tecnologías que más ha avanzado en los últimos años, los teléfonos móviles. Se han vendido en 2014 en el mundo 1000 millones de *smartphones*, además de otros dispositivos que están en auge en la actualidad como son las *tablets*. Para los usuarios a los que se enfoca este proyecto, los avances tecnológicos suponen una notable ayuda en favor de su independencia. Gracias a diferentes sistemas y herramientas, personas con diversidad funcional, se pueden integrar y adaptar con más facilidad al entorno laboral y social.

Dentro del mercado de los dispositivos móviles, nos encontramos con diferentes plataformas como iOS (S.O. de Apple), Android (S.O. de Google) o Windows Phone (S.O. de Microsoft). El sistema elegido para el desarrollo del sistema ha sido Android por su gran cuota de mercado.

Este proyecto nació con la intención de ayudar a personas con diversidad funcional intelectual a guiarse por entornos interiores. “AssisT-In”, forma parte de un proyecto global donde se tiene como objetivo ayudar a este colectivo. Este está formado por diferentes proyectos como “AssisT-Out” (aplicación cuyo principal objetivo es que el usuario consiga llegar a su destino elegido desde un punto de origen, en espacios exteriores) o “AssisT-Task” (proyecto que pretende ayudar a estas personas en la realización de tareas cotidianas como preparar un café o poner una lavadora).

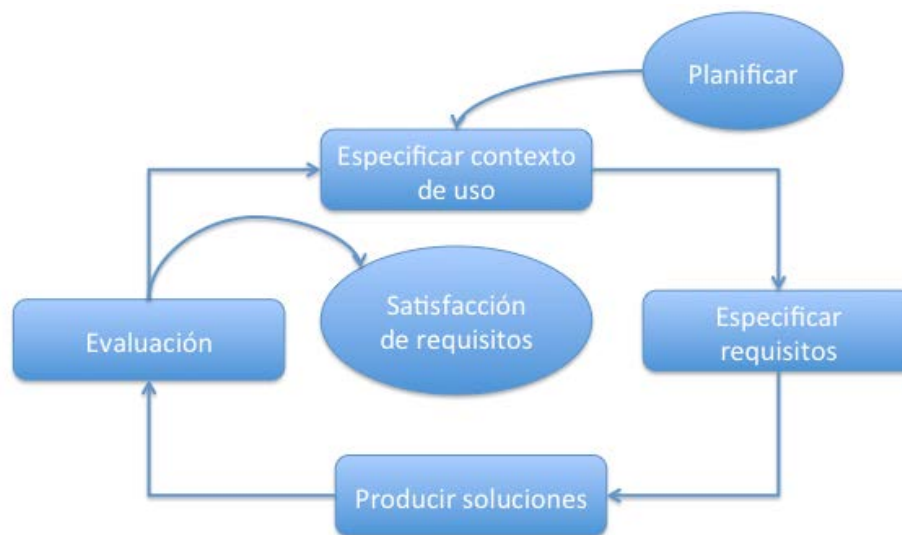
Para la realización del diseño del sistema, puesto que se dirige a un colectivo específico, y en el que se pretende ayudar todo lo posible al usuario, se ha elegido un diseño iterativo y asistido por expertos.

En [27] se comenzó a utilizar el término de diseño centrado en el usuario. Esta filosofía tiene como objetivo principal crear sistemas capaces de resolver las necesidades del usuario, y mejorar su experiencia de uso sin requerir de un esfuerzo extra por su parte. En este proyecto, dadas las características del usuario final, la realimentación se basa en la información proporcionada por el tutor. La norma [28] divide este proceso en las siguientes fases:

1. **Entender y especificar el contexto de uso:** Esto es, conocer a los usuarios finales de la aplicación y el uso que darán al sistema. En relación con este trabajo, se trata de conocer al usuario con diversidad funcional intelectual así como el contexto de uso en el que utilizará el sistema. Este contexto será el día a día del usuario, tratando de facilitarle la tarea de conseguir llegar al destino

elegido. En función de los resultados obtenidos se irá adaptando la aplicación a los usuarios hasta conseguir el objetivo de ir dotando de independencia a dichos usuarios en la tarea de llegar al destino.

2. **Especificación de los requisitos:** Identificar los objetivos del usuario. Para este proyecto esta especificación quedará a cargo de los tutores y educadores de los usuarios, dado que estos no aportarían la información suficiente para esto. Los tutores irán analizando los datos registrados del usuario, y así podrán analizar su evolución y ayudar en posibles mejoras proporcionando esta información.
3. **Crear soluciones posibles:** En función de las diferentes versiones, se puede realizar en varias etapas. Relacionándolo con este trabajo, esta fase irá unida a la información obtenida por los tutores de los usuarios.
4. **Evaluación:** Se validan las soluciones del diseño. Es aquí donde se comprueba si se han cumplido los requisitos o se encuentra algún problema, normalmente detectado por el usuario durante las pruebas.



**Ilustración 2: Diseño iterativo**

## **5.2 Orígenes de la aplicación**

Tal y como se ha comentado con anterioridad en este documento, este proyecto ha sido desarrollado en el laboratorio de inteligencia ambiental (AmILab) de la Universidad Autónoma de Madrid. Aquí se ha desarrollado el proyecto “AssisT”, compuesto de diferentes proyectos, entre los que destacamos una versión anterior de este proyecto.

El proyecto “QRumbs” nace con la intención de ayudar a las personas con diversidad funcional intelectual a conseguir llegar a un destino elegido en entornos interiores. En este

sistema, se contaba con un asistente virtual que iba guiando al usuario. Este mostraba pistas y mensajes durante su transcurso de la ruta a seguir. En la Ilustración 3: Ubicación local en QRumbs, se puede ver dicho asistente. Se trata de una mascota que va guiando al usuario mostrando una serie de informaciones para ir pasando por localizaciones intermedias hasta llegar al objetivo final.



**Ilustración 3: Ubicación local en QRumbs**

En este proyecto se ha decidido eliminar el asistente ya que se ha comprobado que puede dar lugar a confusión a los usuarios de la aplicación. En su lugar se ha añadido un asistente que va guiando al usuario mediante pistas auditivas. Además de mostrar en pantalla la información necesaria para que el usuario consiga su objetivo, se le va guiando mediante voz para una más fácil comprensión.



**Ilustración 4: Imágenes QRumbs**

Otra diferencia respecto al proyecto “QRumbs” es el cambio de diferentes elementos de la interfaz gráfica de la aplicación. A continuación se detallan varias de las pantallas de esta aplicación:

- El “botón del pánico” o botón de ayuda, como se puede visualizar en las maquetas de la aplicación en la Ilustración 4: Imágenes QRumbs, ha sido eliminado en el proyecto. En “QRumbs” este botón estaba disponible en todas las pantallas de la aplicación por si el usuario lo necesitara. Se ha comprobado con el uso de esta aplicación que confunde al usuario, y le puede inducir a situaciones de nerviosismo innecesario. Por esta razón se ha decidido prescindir de un botón de estas características en la aplicación.
- En “QRumbs”, se mostraban los diferentes usuarios disponibles destacando uno, y con dos botones para ir avanzando o retrocediendo. Cuando el usuario veía su foto en pantalla seleccionaba el botón siguiente para avanzar. Esto se puede ver en la Ilustración 4: Imágenes QRumbs. En este proyecto, únicamente se muestra una imagen al usuario, hasta que selecciona el botón OK.
- Las indicaciones que se mostraban en “QRumbs” para localizar el siguiente punto de la ruta se basaban en flechas que el usuario debía ir siguiendo. Esto en muchas ocasiones, puede provocar confusión del usuario a la hora de interpretar en qué dirección tiene que moverse. En nuestro caso se ha decidido mostrar una imagen del siguiente destino que el usuario debe visitar desde el punto en el que se encuentra el usuario. De esta forma será más fácil su localización.
- En la pantalla de elección de destino de “QRumbs” el usuario tenía que hacer una selección de igual forma que en la de usuario, como se ve en la Ilustración 4: Imágenes QRumbs. En este proyecto se muestran todos los destinos en forma de lista, tal y como se observa en la Ilustración 7: Lista de destinos disponibles. El usuario puede ver una imagen y un nombre identificativo por cada destino disponible.
- En esta aplicación además se añaden las pantallas intermedias entre un punto y el siguiente. Cuando el usuario escanea un código QR, el sistema le informa si se ha confundido mediante una pantalla con un pictograma para una fácil comprensión, además de ser informado por el asistente de voz.

Todos estos cambios surgen de la necesidad de unificar la interfaz de los diferentes proyectos desarrollados en AmILab. Se pretende que esta sea uniforme en todos los módulos de AssisT. De esta forma, se conseguirá una transparencia para el usuario en el uso de las aplicaciones, es decir, se ayuda a los usuarios en el paso de una aplicación a otra.



## 5.3 Análisis de requisitos

En todo proyecto software hay una fase de especificación de requisitos. Estos se dividen en funcionales y no funcionales. Los requisitos necesarios para cumplir con los objetivos de este TFG son los siguientes.

### 5.3.1 Requisitos Funcionales

Este tipo de requisitos define las funciones de la aplicación:

- **Conexión a internet:** La aplicación necesitará de conexión a internet para su funcionamiento. No es estrictamente necesaria, aunque sí en su primer uso, puesto que de no haber conexión el sistema utilizará la base de datos descargada en el *smartphone*. Al tratarse de un diseño iterativo e incremental, la aplicación enviará un correo al finalizar el usuario, enviando información sobre su uso. Esto ayudará a la realimentación.
- **Sistema Operativo Android:** Es estrictamente necesario que el dispositivo móvil disponga del sistema operativo Android instalado.
- **Dispositivo móvil con aplicación de cámara de fotos:** Aunque hoy en día prácticamente todos los *smartphones* disponen de cámara de fotos, es un requisito imprescindible. Las pistas que se dan al usuario para conseguir su objetivo se realizan mediante la captura de códigos QR, por lo que es necesario que el dispositivo móvil disponga de aplicación de cámara de fotos mediante la cual se hará el escaneo.
- **Aplicación captura QR:** La captura de códigos QR se hará mediante la aplicación BarCode Scanner. El propio sistema comprobará si se tiene instalada esta aplicación en el dispositivo y, en caso negativo, se instalará automáticamente.

### 5.3.2 Requisitos No funcionales

También llamados atributos de calidad de un proyecto software, son aquellos que no describen información ni funciones del sistema:

- **Bajo coste:** Puesto que se tiene la intención de poder implantar el sistema en multitud de entornos diferentes, el coste debe ser bajo. Para la implantación inicial del sistema en un único entorno solo se necesitará del dispositivo móvil del usuario y de la señalización de puntos clave mediante etiquetas QR, lo que hace que el coste no sea elevado.
- **Interfaz de usuario sencilla:** El colectivo al que se dirige esta aplicación requiere que la interfaz de usuario sea lo más sencilla posible. Además se requiere que toda la información se muestre en la pantalla del teléfono del

usuario independientemente de su tamaño. La información se mostrará en dos versiones: gráfica y auditiva, por lo que debe ser también sencilla en la medida de lo posible para una mejor comprensión.

- **Diseño iterativo e incremental** : Para una realimentación adecuada, se hará partícipe al usuario de las mejoras en la aplicación. Los tutores serán los encargados de ir proporcionando la información necesaria para posibles mejoras y solución de errores.
- **Robustez o estabilidad**: Este requisito es uno de las más importantes del proyecto. Puesto que se trabaja con usuarios con diversidad funcional intelectual, no se pueden dar errores en el sistema que le hagan confundirse. Ya que se tiene como objetivo que vaya ganando independencia a medida que va usando la aplicación, un fallo en el sistema puede dar lugar a confusión y una pérdida de confianza.

## 5.4 Diseño

El uso de un dispositivo móvil de los existentes hoy en día, con la capacidad y características de estos hacen posible este tipo de sistemas. Estas características van mejorando a una enorme velocidad, dotando a la mayoría de los dispositivos de las últimas tecnologías desarrolladas. Con estos *smartphones*, se puede acceder desde cualquier lugar a la red gracias a sus posibilidades de comunicación.

En este proyecto, el acceso a la información recae íntegramente en los tutores y educadores de los usuarios. Esto es así, ya que se garantiza que estos puedan proporcionar con mayor fiabilidad y certeza aquella información que pueda ser útil para la solución de posibles errores o mejoras del sistema.

La función de los usuarios se limita únicamente al uso de la aplicación. El sistema irá guardando la información que estos proporcionen con el uso del sistema, para que los educadores puedan analizar posteriormente los resultados.

Para el desarrollo de la aplicación se utilizará la programación orientada a objetos. Más concretamente, y como ya se ha comentado antes, se programará en Android, que es el sistema operativo requerido por los dispositivos móviles que usarán la aplicación.

### 5.4.1 Diseño de la interfaz

En este apartado del capítulo se mostrarán las maquetas iniciales de la aplicación. Dichas maquetas se han realizado con la herramienta Balsamiq Mockups, que proporciona los elementos necesarios para construirlas.

El principal objetivo en el diseño de la interfaz es la sencillez que se debe dar a la aplicación. Debe ser también lo más intuitiva posible. Además, deben ser lo

suficientemente completas como para cumplir los objetivos que nos propusimos en los Objetivos del proyecto.

La aplicación al comenzar muestra una pantalla de espera al usuario. Se puede ver en la Ilustración 5: Pantalla de carga de AssisT-In. Durante esta pantalla de espera la aplicación internamente hace la conexión con el servidor en caso de haber conexión a la red, y se descarga la última versión de la base de datos disponible en dicho servidor.



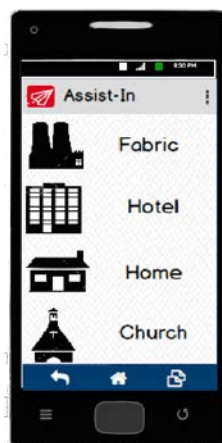
**Ilustración 5: Pantalla de carga de AssisT-In**

Una vez la aplicación ha arrancado, se le mostrará al usuario una imagen por cada usuario de la base de datos. Tiene a su disposición dos botones que cambiarán la imagen del usuario mostrada. Una vez el usuario ha encontrado su foto, seleccionará el botón OK para hacer su selección. Esta pantalla se ve en la Ilustración 6: Lista de usuarios y búsqueda de código QR.



**Ilustración 6: Lista de usuarios y búsqueda de código QR**

Con el usuario elegido, el sistema le informará que debe buscar en su alrededor una pista como la que se muestra en la Ilustración 6: Lista de usuarios y búsqueda de código QR. En este momento deberá escanear el código QR encontrado para ubicar su situación.



**Ilustración 7: Lista de destinos disponibles**

Una vez el sistema conoce la ubicación del usuario le pedirá que seleccione el destino al que se quiere dirigir, tal y como se ve en la Ilustración 7: Lista de destinos disponibles. La información se le muestra en forma de lista con una imagen a la izquierda del nombre del destino posible. De esta forma se hace más sencilla la comprensión para el usuario.

Cuando elige el destino al que se quiere dirigir, el sistema calcula la ruta y le empieza a mostrar al usuario las indicaciones que debe ir siguiendo. Una de estas indicaciones se puede ver en la Ilustración 8: Búsqueda de destino.



**Ilustración 8: Búsqueda de destino**

Las pistas que se le muestran al usuario, vienen complementadas con dos botones para que el usuario indique si desea buscar la pista pedida o volver al anterior punto en caso de haberse perdido. Al realizar la implantación del sistema de etiquetas QR, cada destino final o intermedio del entorno se marcará con una de estas etiquetas. En las imágenes que se mostrarán al usuario, dicha etiqueta saldrá marcada con el borde en color rojo para que la identificación por parte del usuario a la hora de buscar la pista en el destino sea más rápida.

Cuando el usuario se disponga a escanear el código QR que la aplicación le ha pedido, el sistema abrirá la aplicación elegida para el escaneo del código QR BarCode Scanner. Con la pista ya escaneada, el sistema mostrará por pantalla si la pista se corresponde con el siguiente punto de la ruta al que se debía dirigir o por el contrario se ha confundido. En caso de que el usuario haya llegado al siguiente punto, el sistema mostrará una pantalla como la Ilustración 9: Pista correcta.



**Ilustración 9: Pista correcta**

En caso de haberse equivocado, el sistema indicará si se ha escaneado una etiqueta QR que no pertenece a la ruta o si se ha saltado alguna pista. Estas 2 opciones se observa en la Ilustración 10: Pistas incorrectas.



**Ilustración 10: Pistas incorrectas**

La elección de pictogramas para estas pantallas de la aplicación, obtenidas del portal del gobierno de Aragón [29], se ha realizado teniendo en cuenta las características del público al que va dirigida la aplicación. Es importante reforzar la motivación de este tipo de usuario al realizar un paso correctamente, así como indicar mediante estos pictogramas cuando se ha equivocado.

Estas pantallas se irán sucediendo mientras el usuario vaya persiguiendo las pistas que el sistema le va mostrando. Una vez el usuario llega al destino que había seleccionado al principio, la aplicación le muestra que ha concluido con éxito, como se ve en la Ilustración 11: Destino correcto.



**Ilustración 11: Destino correcto**

La aplicación termina y se cierra cuando el usuario selecciona el botón de esta última pantalla.

# 6 Implementación

---

## 6.1 Arquitectura del sistema

Este proyecto forma parte de los proyectos desarrollados en el laboratorio AmILab de la Universidad Autónoma de Madrid. La arquitectura propuesta en este sistema es cliente-servidor. En el servidor se incluye la base de datos necesaria, donde se hace uso de otras tablas compartidas con los diferentes proyectos de este laboratorio. La arquitectura se muestra de forma esquemática en la Ilustración 12: Arquitectura cliente-servidor. Se puede observar en esta imagen, que el cliente comienza la conexión desde la aplicación Android con el servidor, dónde se encuentra la base de datos. En caso de disponer de conexión a internet el dispositivo del usuario, el servidor responderá con la última versión de la base de datos alojada.

A continuación se detallan las características de este sistema por separado.

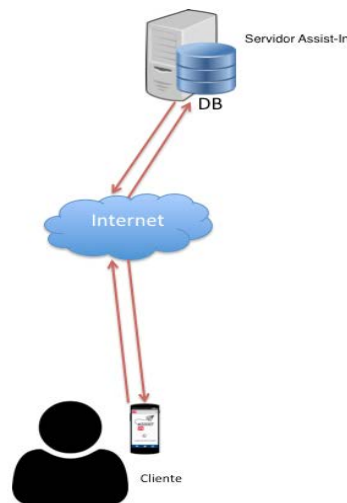


Ilustración 12: Arquitectura cliente-servidor

## 6.2 Servidor

Como ya se ha comentado anteriormente, el papel que juega el servidor en este proyecto es muy sencillo. El cliente realizará una petición mediante llamadas http al servidor alojado en el laboratorio AmILab con el fin de obtener la última versión de la base de datos disponible de la aplicación.

La principal ventaja de esto, es que no se requiere de manera imprescindible conexión a internet para el uso de la aplicación. Ya que la implantación del sistema se realizará en

entornos interiores, es muy común que en determinados momentos el dispositivo móvil del usuario pierda la conexión. En este caso, la aplicación dejaría de funcionar.

El servidor del sistema es el encargado de almacenar esta base de datos, así como las tareas de gestión. Serán los tutores y educadores de los usuarios quienes gestionen la información almacenada. Cuando el cliente, mediante el inicio de la aplicación por parte del usuario, realiza la petición al servidor, se establece la conexión y se procede a la descarga la base de datos en el dispositivo. En el apartado 6.3.2 se explica en detalle en qué módulos y cómo se realiza esta conexión.

Como ya se ha mencionado, este proyecto comparte recursos con otros desarrollados en el laboratorio AmILab. El servidor forma parte de estos recursos compartidos.

Para la tarea de descarga de imágenes desde el servidor, se requiere de diferentes ficheros php, alojados en el servidor. Estos ficheros son:

- ***dump.php***: conecta con la base de datos del servidor. Además ejecuta el script “*mysql2sqlite.sh*”, también alojado en el servidor, cuya función es la conversión de la base de datos existente en formato “mysql”, al formato admitido por los dispositivos móviles con el sistema operativo Android, es decir, “sqlite”.
- ***getImage.php***: para la descarga de imágenes (usuarios y destinos).
- ***simpleImage.php***: formatea en servidor el tamaño de las imágenes descargadas.

El servidor “AssisT-In” es también el encargado de almacenar el registro de actividad de los usuarios de la aplicación. Gracias a estos registros, los tutores pueden analizar la actividad de los usuarios para posibles mejoras. El registro se envía al servidor por correo electrónico una vez el usuario ha finalizado el uso de la aplicación.

## **6.3 Cliente**

En este apartado del capítulo se explican detalladamente los módulos, o paquetes, de la aplicación Android.

### **6.3.1 Estructura del cliente**

Los elementos que componen el cliente son:

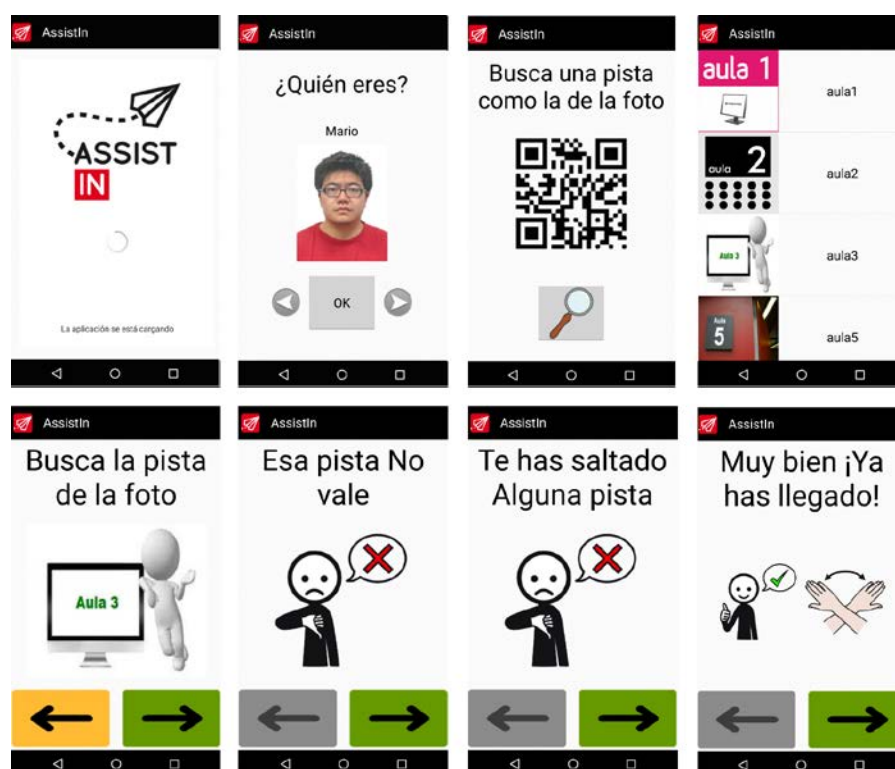
1. **Dispositivo móvil**: El dispositivo que emplea el usuario para el uso de la aplicación, y todos sus elementos (cámara de fotos, aplicación externa BarCode Scanner, sonido, etc.) forman parte del cliente.



2. **Las diferentes pantallas de la aplicación:** Las pantallas mostradas en las maquetas anteriormente forman una parte importante del cliente. Estas pantallas son:
- a. *Elección de usuario:* El usuario de la aplicación buscará su imagen en el dispositivo. Irá avanzando entre imágenes mediante los botones de la aplicación para avanzar o retroceder. Una vez vea su imagen en el dispositivo, seleccionará el botón Ok para hacer su selección.
  - b. *Elección de origen:* La elección del origen en el sistema se hace de la siguiente manera. La aplicación muestra al usuario que debe buscar en su alrededor una pista como la que se ve en la imagen del dispositivo. Esta imagen es una etiqueta QR, mediante la cual el sistema identificará donde se encuentra el usuario. Las etiquetas QR se componen de un texto tal como “**Assist-in:mXXXnYYY**”. En este texto se identifica el nombre de la aplicación “Assist-In”, seguido de los datos que necesita el sistema para ubicar al usuario en el entorno. Estos datos son el número de mapa y número de nodo. En este caso, el mapa se identifica mediante “mXXX”, y el nodo o punto del entorno, mediante “nYYY”. Por ejemplo, una etiqueta QR válida es “Assist-In:m002n003”, que equivale al nodo 3 del mapa 2.
  - c. *Escáner:* Cuando el usuario ha localizado la pista que le ha proporcionado el sistema para buscar, lanzará la aplicación BarCode Scanner. Esta será la encargada de escanear el código QR que identifica el origen del usuario.
  - d. *Elección de destino:* Con el código QR escaneado en el punto de origen del usuario, el sistema le pedirá al usuario que seleccione el destino al que se quiere dirigir. La información se muestra mediante una lista en la que por cada destino posible se ve una imagen identificativa del destino, y el nombre de dicho destino. Dado que la información debe verse correctamente en el dispositivo, y requiere de cierto tamaño, en esta pantalla se dispone de *scroll-bar* para mostrar toda la lista de destinos. El usuario deberá bajar o subir en la pantalla del dispositivo para localizar el destino.
  - e. *Pistas:* Una vez el usuario ha elegido su destino, el sistema internamente calculará la ruta. Más adelante se explicará cómo se realiza este cálculo. Con la ruta calculada, se le irá mostrando al usuario una serie de pistas que deberá ir recorriendo para llegar a su objetivo final. Las pistas que se muestran son, tal y como se indica en la maqueta de la elección de destino, mediante una imagen remarcando la etiqueta QR dispuesta en ese punto. El usuario después de escanear la pista pedida por la aplicación obtendrá un resultado.

f. *Mensajes resultado de las pistas:* Hay 4 tipos de mensajes que muestran al usuario tras escanear una de las pistas de la ruta. Estos mensajes se muestran en función del resultado de este escaneo. Si el usuario ha escaneado la etiqueta QR del siguiente punto de la ruta que debía visitar, se le mostrará un mensaje indicando que ha elegido correctamente. En caso de que el usuario haya escaneado un código QR equivocado, pero perteneciente a la ruta que debe seguir, la aplicación le indicará que se ha saltado alguna pista. Si se encuentra con un código QR ajeno a nuestra aplicación, el sistema le indicará que se trata de una pista incorrecta. En estos dos últimos casos, se le pedirá al usuario que vuelva a buscar la pista. Por último, si el usuario escanea la etiqueta QR que identifica el destino final elegido, el sistema mostrará que ha llegado al final de su ruta de forma correcta.

A continuación, en la Ilustración 13: Imágenes finales de la aplicación, se muestran las imágenes de la aplicación final, dónde se pueden ver las pantallas citadas anteriormente:

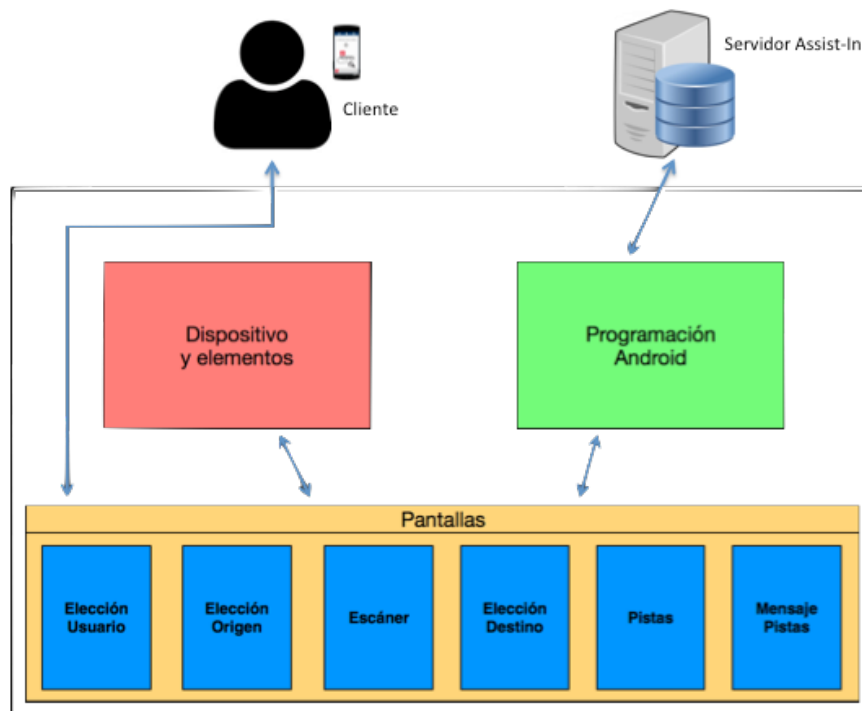


**Ilustración 13: Imágenes finales de la aplicación**

3. **Programación de la aplicación:** La lógica programada en Android de la aplicación es una parte fundamental del cliente del sistema. Es donde se realiza la comunicación con el servidor, así como la encargada de gestionar la transición entre las diferentes pantallas mencionadas anteriormente. Además,

se realiza el cálculo de la ruta según el algoritmo BFS de búsqueda en anchura para el recorrido de grafos.

A continuación se muestra de forma esquemática la estructura de la parte cliente del sistema.



**Ilustración 14: Estructura cliente AssisT-In**

### 6.3.2 Módulos de la aplicación

#### Comunicaciones

Esta aplicación se basa en una arquitectura cliente-servidor, donde la parte cliente es nuestra aplicación para dispositivos móviles.

En todo proyecto software con esta arquitectura se realizan comunicaciones con el servidor que ofrece una serie de servicios al cliente. En este caso, estas comunicaciones se realizan en este módulo de comunicaciones

Este módulo lo componen dos clases Java:

- Mail.java
- RouteInfoProvider.java

### ***RouteInfoProvider.java***

En esta clase se realiza la gestión de descarga de la base de datos desde el servidor. Como ya se ha comentado con anterioridad, la interacción con el servidor es muy simple.

Únicamente se realiza una petición al servidor, la descarga de la base de datos en caso de tener conexión (WiFi, HDSPA, etc.,). Para ello tenemos la clase *RouteInfoProvider.java*.

```
public RouteInfoProvider(Context, boolean online){
    this.context = context;
    /* Si hay conexión a Internet,
     * descargamos toda la base de datos del servidor */
    if(online){
        /* Eliminar base de datos previa */
        clear();
        /* Lanzar tarea de descarga */
        new RetrievalTask(context).execute();
    } else {
        /* Modo offline: abrir base de datos precargada */
        init();
        /* Notificamos a la actividad llamadora que
         * se ha finalizado la inicialización */
        ((BaseActivity) context).providerCallback();
    }
}
```

#### **Ilustración 15: Constructor RouteInfoProvider**

El constructor de esta clase comprueba si existe conexión a internet en el dispositivo del usuario. En caso de que disponga de conexión se lanza la tarea de descarga de la base de datos desde el servidor de AmILab. Antes de la descarga de la base de datos, se elimina la última versión de esta base de datos disponible en el dispositivo para tenerla actualizada.

Esta tarea de descarga es una tarea asíncrona que extiende de AsyncTask con el objetivo de que la aplicación no se bloquee durante la descarga.

La descarga de la base de datos del servidor incluye la descarga de las imágenes requeridas por la aplicación. Estas imágenes son las fotos de los usuarios disponibles, así como de los destinos a los que el usuario se podrá dirigir.

Esta descarga de imágenes se realiza mediante dos clases internas ejecutadas como tareas asíncronas, llamadas *UserImagesDownloader* y *DestinationImagesDownloader*. En estas, se crea una carpeta en el dispositivo, donde se guardarán las imágenes descargadas. A continuación, mediante una petición http se descargarán dichas imágenes. Se muestra a modo de ejemplo el código descrito de *UserImagesDownloader*, pero en ambas clases se hace de igual manera.

```

/* Obtenemos el path para escribir en la tarjeta SD */
File dirFile = new File(Environment.getExternalStoragePublicDirectory
(Environment.DIRECTORY_PICTURES).toString() + PATH);
dirFile.mkdirs();
for(String photo: users.values()){
    OutputStream outStream = null;
    File file = new File(dirFile.getAbsolutePath(), photo);
    try {
        outStream = new FileOutputStream(file);
        String url = "http://" + Preferences.getIP(context) +
"/AssistIn/getImage.php?img=user_data/Photos/"+ photo + "&w=55&h=200";
        Log.d("assistentin.db", "Downloading photo " + url + " to " +
dirFile.getAbsolutePath());
        copyStream(openHttpConnection(url), outStream);
        outStream.flush();
        outStream.close();
    }
}

```

**Ilustración 16: Descarga de imágenes UserImagesDownloader**

### ***Mail.java***

La otra clase que forma el módulo de comunicaciones de la aplicación es *Mail.java*. Se hace uso de esta clase una vez el usuario ha terminado su uso. Durante el transcurso de la aplicación se va guardando un registro de la actividad del usuario con la aplicación, para finalmente ser enviado por correo electrónico al servidor AmILab.

### **Base de datos**

Este módulo lo forman dos clases Java: *DatabaseAdapter.java* y *DatabaseFeed.java*. Toda la gestión relacionada con la base de datos en la aplicación se realiza en este módulo.

### ***DatabaseFeed.java***

Dada la arquitectura del sistema, cliente-servidor, la interacción con el servidor se limita a la descarga de la base de datos disponible. Esta comunicación se realiza en *DatabaseFeed.java*

Mediante una llamada http desde la aplicación cliente se hace la petición al servidor, que responde sin errores en la descarga si todo ha ido correctamente.

```

url = url.replace(" ", "%20");
HttpClient client = new DefaultHttpClient();
HttpGet httpGet = new HttpGet(url);
String result = "";
try {
    HttpResponse response = client.execute(httpGet);
    StatusLine statusLine = response.getStatusLine();
    int statusCode = statusLine.getStatusCode();
    if (statusCode == 200){
        HttpEntity entity = response.getEntity();
        result = EntityUtils.toString( entity, HTTP.UTF_8);
        Log.d("DBFEED", result);
    } else {
        Log.e("DBFEED", "Failed to download database");
    }
}

```

**Ilustración 17: Conexión con servidor**

## ***DatabaseAdapter.java***

*DatabaseAdapter.java* cuenta con una clase interna cuyo único objetivo es crear la base de datos y obtener un objeto para representarla. Es sobre este objeto sobre el que se realizarán todas las consultas necesarias a la base de datos. Esta clase interna recibe el nombre de *DatabaseHelper*.

En esta clase se hace la gestión principal de la base de datos. Se crea, se abre y se cierra cuando sea necesario.

En la clase principal *DatabaseAdapter.java* se construyen las consultas SQL que utilizará la aplicación para obtener la información de la base de datos. Más adelante se explicará un ejemplo de uso de la aplicación, y con ello, las consultas que realiza la aplicación.

## **Utils**

Este es un módulo con diferentes clases que se utilizarán en la aplicación para objetivos específicos. Es el caso de *AssistLog.java*, *IntentIntegrator.java* e *IntentResult.java*.

## ***AssistLog.java***

Se trata de una clase usada en los diferentes proyectos desarrollados en AmILab y citados en el capítulo 5. En nuestra aplicación Android, se va guardando un registro de la actividad que va realizando el usuario. Esta información es útil, para que el tutor conozca su comportamiento durante el transcurso de la ruta. Esta información, forma una parte importante del modelo de proceso iterativo seguido en este proyecto, ya que proporciona información al tutor del usuario y este puede decidir si requiere de alguna modificación en la ruta, o si algún elemento de la aplicación debe ser modificado a favor del usuario.

A continuación se detalla que tipo de información es registrada.

```
public static enum LEVEL {ERROR, INFO, SERVER, WARNING};  
  
public static enum ACTION_TYPE {CONFIG, INTERACTION, GENERAL, EXECUTION};
```

### **Ilustración 18: Tipo de información registro de actividad**

Se tienen dos enumerados como se ve en la imagen anterior. El primero de ellos denominado *LEVEL* clasifica la información que se va a registrar:

- **ERROR:** Informa de que se ha producido algún tipo de error en la aplicación mientras el usuario hace uso de ella. Por ejemplo, si la descarga de la base de datos no finaliza de forma correcta.

- **INFO:** Es el nivel más utilizado en el registro de actividad. Proporciona información de carácter general, como puede ser el nombre del usuario del que se va a registrar la actividad, el punto de origen de la ruta o el destino elegido.
- **SERVER:** Este nivel de registro da información proporcionada por el servidor. Esto no se utiliza dada la poca interacción entre cliente y servidor.
- **WARNING:** Nivel de registro poco utilizado. Da información de menor importancia.

El segundo enumerado de la clase AssistLog.java es *ACTION\_TYPE*, que proporciona la información en función del tipo de acción realizada.

- **CONFIG:** Se registra una acción de configuración por parte del usuario. Se utiliza, por ejemplo, en la inicialización de la aplicación para informar de que se ha recuperado toda la información de la base de datos.
- **INTERACTION:** Se trata de una acción realizada directamente por el usuario. Se registra mediante este tipo, información como la elección de usuario o destino.
- **GENERAL:** Con este tipo de acción se registra la mayoría de la información. Se registra cual es el origen del usuario, su destino, los puntos de la ruta por los que va pasando, etc.
- **EXECUTION:** No se utiliza este tipo de acción en.

### ***IntentIntegrator.java***

Cuando el usuario de la aplicación procede a identificar donde se encuentra, o a buscar una pista, se le pide escanear una etiqueta QR disponible en el entorno. Para poder escanear estos códigos QR, el sistema requiere de una aplicación. La aplicación en cuestión es BarCode Scanner.

El sistema facilitará la función del escaneo por parte del usuario al proporcionar un enlace directo a la descarga de esta aplicación de terceros. La clase encargada de esta gestión es *IntentIntegrator.java*.

### ***IntentResult.java***

Relacionada de forma directa con la clase anterior tenemos *IntentResult.java*. Esta clase está compuesta por el propio contenido de la etiqueta QR leída y de un nombre.

Cuando desde la aplicación se quiere identificar el código, se requiere de esta información.

Como ejemplo de uso se muestra a continuación una llamada al método *getContents()*:

```
IntentResult scanResult = IntentIntegrator.parseActivityResult  
(requestCode, resultCode, data);  
if (scanResult != null) {  
    _scannedEntity= scanResult.getContents();  
    scanned = true;  
}
```

#### **Ilustración 19: Método getContents**

### **Actividades**

El módulo de actividades forma el núcleo de nuestra aplicación Android. Se compone de las siguientes clases Java:

1. BaseActivity.java
2. BFS.java
3. DestinationSelectorActivity.java
4. FinishActivity.java
5. MapActivity.java
6. MessageQRActivity.java
7. OriginSelectorActivity.java
8. Preferences.java
9. QR.java
10. UserSelectorActivity.java

#### **6.3.3 Paso a paso entre actividades**

A continuación se explicará en detalle el módulo de actividades simulando el uso de la aplicación por parte de un usuario.

Cuando el usuario inicia la aplicación, el sistema comienza con la clase *BaseActivity.java*. Esta clase es la encargada de gestionar el comienzo de la aplicación. Aquí, se configuran los datos iniciales como el tutor del usuario que va a usar el sistema, o donde se comprueba si existe conexión a internet.

El método *checkConex()* comprueba todas las redes del dispositivo donde se usa la aplicación, y devuelve true en caso de tener disponible alguna de las redes.



```

public static boolean checkConec(Context ctx){
    ConnectivityManager connec =
    (ConnectivityManager)ctx.getSystemService(Context.CONNECTIVITY_SERVICE);
    /* Con esto recogemos todas las redes que tiene el móvil (wifi, gprs...)
    */
    NetworkInfo[] redes = connec.getAllNetworkInfo();
    for(int i=0; i<redes.length; i++){
        if (redes[i].getState() == NetworkInfo.State.CONNECTED){
            return true;
        }
    }
    return false;
}

```

### Ilustración 20: Método checkConex

Desde esta clase *BaseActivity*, se gestiona la cadena de llamadas entre diferentes actividades. Esta gestión se realiza en el método *onActivityResult()*, donde se interpreta el resultado de una actividad para iniciar la siguiente.

Cuando el sistema arranca se le pide al usuario que seleccione su imagen para ser identificado. Se inicia así la actividad *UserSelectorActivity.java*. En esta clase se gestiona toda la actividad del usuario. Para que el usuario pueda identificarse se le muestran las imágenes de todos los usuarios dados de alta en el sistema, como se ve en Ilustración 6: Lista de usuarios y búsqueda de código QR. El método *updateScreen()* es el encargado de esto, cargando las imágenes descargadas en el dispositivo móvil.

Una vez el usuario ha seleccionado su foto, queda identificado por el sistema y el control pasa de nuevo a *BaseActivity.java*. Desde aquí se lanza la actividad que da opción a situar al usuario, *OriginSelectorActivity.java*. En esta clase se le pide que busque un código QR como el que se le muestra de pista. Cuando el usuario localiza a su alrededor una etiqueta QR, debe seleccionar el botón “Lupa” para escanear dicha etiqueta.

El sistema ya tiene identificado quién es el usuario de la aplicación, así como su punto de origen en el entorno. Ahora el control vuelve de nuevo a *BaseActivity*, para dar paso a la actividad *DestinationSelectorActivity.java*. Tal y como se muestra en Ilustración 7: Lista de destinos disponibles, se proporciona al usuario una lista de los destinos disponibles para elegir. Cabe destacar, que no todos los destinos de un mapa son seleccionables como destino final del usuario. Estos destinos seleccionables están identificados en la base de datos con el atributo “*nodeFinal=1*”, como se explicará en el capítulo Base de datos. De cargar los datos (imagen y nombre) de los destinos que podrá seleccionar el usuario se encarga el método *inflateDestinationList()*.

```

private void inflateDestinationList() {
    Map<String, String> destinationmap =
    BaseActivity.provider.getDestinations(idMap);
    destinations = new ArrayList<String>(destinationmap.keySet());
    photos = new ArrayList<String>(destinationmap.values());
}

```

### Ilustración 21: Método inflateDestinationList

Una vez el usuario ha seleccionado en la lista disponible a que destino se quiere dirigir, el sistema guarda esta selección. Con el punto de origen, el punto de destino y el usuario que está haciendo uso de la aplicación, se calcula la ruta que le llevará hasta su objetivo.

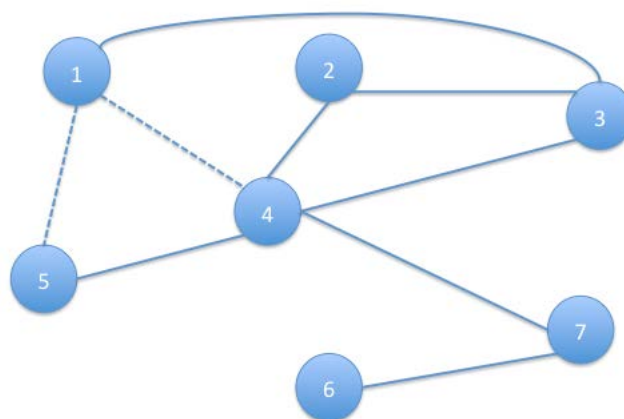
Para el cálculo de esta ruta se construye un objeto BFS, como se muestra a continuación, de la clase *BFS.java*.

```
int [][] grafo = generaGrafo();  
  
BFS bfs = new BFS(grafo);
```

#### **Ilustración 22: Llamada a método generaGrafo**

El grafo generado para el cálculo de la ruta se crea a partir de la tabla “*in\_relationNodes*” de la base de datos. Esta tabla guarda las relaciones que existen entre los diferentes nodos de un mapa. Con los datos necesarios, es decir, punto de origen, punto de destino y usuario, se procede al cálculo de la ruta que debe seguir para conseguir llegar a su destino elegido.

Es importante destacar como se forma el grafo. Se ha decidido que los tutores pueden introducir variaciones específicas en la ruta que debe seguir un usuario, basándose en su experiencia. Estas variaciones tienen la única intención de facilitar la ruta. Con variación de ruta se considera la conexión entre dos nodos. Por ejemplo, para ir desde el punto 1 al punto 4 un usuario sin especificaciones en su ruta por parte del tutor, tendría que pasar por el punto 2 y a continuación dirigirse al punto 4. En cambio, para un usuario cuyo educador ha considerado que sería más sencillo para él atravesar directamente desde el origen en el punto 1 al punto 4, se establecería con mayor prioridad esta conexión que la de un usuario sin especificaciones.



#### **Ilustración 23: Grafo representativo de mapa de AssisT-In**

A la hora de formar el grafo, una conexión normal en el grafo se establece con valor “1” y una conexión específica mediante el valor “3”. Es importante señalar que estas

consideraciones tienen mayor prioridad que una conexión normal entre dos nodos, siempre y cuando sirva para acortar la ruta del usuario. Un ejemplo de una matriz generada de un grafo se muestra en la Tabla 4: Matriz de un grafo representativo de un mapa. En el siguiente apartado se explica en detalle el algoritmo utilizado.

### ***Algoritmo BFS***

El algoritmo elegido mediante el que se calcula la ruta a seguir por el usuario es BFS (*Breadth First Search*, Búsqueda En Anchura) añadiendo las consideraciones explicadas en el punto anterior. En el Anexo A: Algoritmo Breadth First Search se muestra el pseudocódigo del algoritmo.

Con el grafo bien formado, se llama al método *recorridoAnchura(nodoInicial, nodoFinal)*. Este será el encargado de devolver la lista de nodos o puntos del mapa que el usuario tendrá que ir visitando por pasos desde el origen hasta su destino.

El algoritmo comienza añadiendo el nodo inicial a la cola. Una vez hecho esto, se comprueba en cada paso si la cola está vacía, cuando la cola no tenga ningún nodo, el algoritmo habrá terminado.

En cada iteración del bucle se comprueba si el nodo actual se corresponde con el nodo final. Este nodo actual es el último extraído de la cola. A continuación se recorre el grafo buscando las conexiones específicas del usuario. Como ya se ha comentado, estas conexiones tienen prioridad respecto a una conexión normal entre dos nodos, por lo que son añadidas antes a la cola. Una vez se han añadido todas se vuelve a recorrer el grafo para el resto de uniones.

Se van extrayendo los nodos de la cola en cada iteración del bucle hasta que se llega al nodo final. El método devuelve la lista de nodos a visitar por el usuario.

Con el recorrido de la ruta calculado se vuelve a la actividad *DestinationSelectorActivity.java*, donde se guardan los datos del destino elegido y la propia ruta obtenida en preferencias, necesario para el registro de actividad. Ahora vuelve el control a *BaseActivity* y se lanza la actividad *MapActivity.java*, desde donde se controla la aplicación hasta el final.

*MapActivity* es la actividad encargada de ir mostrando al usuario las pistas que debe ir siguiendo hasta el final de su ruta. Se van mostrando las pistas de una en una, tal y como se observa en la Ilustración 8: Búsqueda de destino. En el método *onActivityResult()* se analiza el resultado devuelto por la actividad anterior. Esto es, si el usuario ha sabido encontrar la pista o se ha equivocado.

En función del resultado, desde *MapActivity* se lanza la actividad *MessageQRActivity.java* con una información u otra. Los posibles resultados son:

- **QR\_OK:** El usuario ha sido capaz de encontrar el siguiente punto de la ruta y ha escaneado correctamente el código QR de la pista. En este caso,

*MessageQRActivity* se lanza informando de que la pista es correcta y se muestra una nueva pista con el siguiente punto que debe visitar.

- **QR\_NOT\_ORDER:** Este resultado indica que el usuario ha leído una pista QR equivocada de la ruta. El sistema le vuelve a mostrar la pista que debe buscar para seguir.
- **QR\_NOT\_ROUTE:** El usuario ha escaneado con su dispositivo una etiqueta QR errónea. No se corresponde con los códigos de la aplicación. De nuevo el sistema le muestra la pista del punto de la ruta que debe visitar.
- **QR\_END:** Indica que el usuario ha encontrado el final de su ruta de forma correcta. La aplicación le muestra que ha llegado a su destino.

Si el usuario se pierde en algún momento de la ruta y desea volver al punto anterior, tiene disponible el botón de atrás, y el sistema volverá a mostrar las indicaciones de ruta desde ese punto.

### ***TextToSpeech***

La aplicación incorpora un sistema de guiado por voz al usuario. Además de las pistas visuales explicadas, se va guiando mediante pistas de audio. En cada pista que se le muestra al usuario para visitar el siguiente punto de su ruta, la aplicación guía al usuario mediante voz, así como una vez leída una etiqueta QR del entorno se informa del resultado.

Se incorpora el *TextToSpeech* en cada actividad de la aplicación, para una mejor interacción con el usuario, lo que facilitará su uso.

La inicialización del *TextToSpeech* se realiza en el método *onInit()* de cada actividad. A continuación se muestra como ejemplo en la actividad de *MapActivity*.

```
@Override
public void onInit(int status) {
    if (status == TextToSpeech.SUCCESS) {
        tts.setLanguage(Locale.getDefault());
        tts.speak("Busca la siguiente pista",
TextToSpeech.QUEUE_FLUSH, null);
    }
}
```

#### **Ilustración 24: Método onInit de TextToSpeech**

Es necesario cerrar el *TextToSpeech* una vez la actividad se cierra. Para ello se invoca el método *onDestroy()* en cada una de las actividades.

```

@Override
protected void onDestroy() {
    super.onDestroy();
    if (tts!=null) {
        tts.stop();
        tts.shutdown();
    }

    AssistLog.addMessage(AssistLog.LEVEL.INFO, Preferences.getUsername(this),
        AssistLog.ACTION_TYPE.GENERAL, "EXIT", "logged out");
    AssistLog.sendLog("Educador@gmail.com", Preferences.getUsername(this));
    AssistLog.saveLog(Preferences.getUsername(this));
    AssistLog.sendLogFile(Preferences.getUsername(this));
}

```

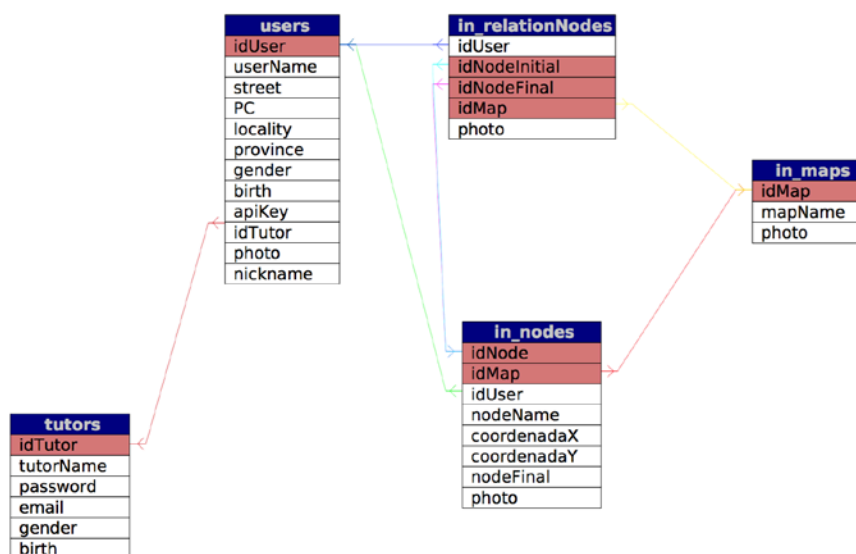
**Ilustración 25: Método onDestroy de TextToSpeech**

Como se puede observar en este método, al finalizar y cerrar la aplicación, se procede a guardar todo el registro de actividad del usuario. Una vez guardado, se envía por correo electrónico el archivo con esta información al servidor de AmILab. Estos archivos se guardan por cada usuario que utiliza la aplicación.

## 6.4 Base de datos

En este apartado del capítulo se explica en detalle el esquema relacional de la base de datos de la Ilustración 26: Esquema relacional de base de datos. Como ya se ha comentado con anterioridad, en el laboratorio AmILab se trabaja en diferentes proyectos. En cuanto al diseño de la base de datos, se trabaja conjuntamente entre los diferentes proyectos.

En particular, para este Trabajo Fin de Grado contamos con dos de las tablas existentes y necesarias para otros proyectos. Estas tablas son “users” y “tutors”. A continuación se muestra de forma esquemática el modelo de base de datos.



**Ilustración 26: Esquema relacional de base de datos**

A continuación se pueden ver en detalle las tablas de “*in\_nodes*”, “*in\_maps*” y “*in\_relationNodes*”.

1. Tabla “*in\_nodes*”: Cabe destacar en esta tabla, que se guarda la información necesaria para la identificación de un nodo de un mapa, teniendo como clave primaria los atributos “*idNode*” y “*idMap*”. Como se ve en la Ilustración 26: Esquema relacional de base de datos, esta tabla está relacionada con la tabla “*users*” mediante el atributo “*idUser*”; con “*in\_Maps*” mediante la PK de esta, es decir “*idMap*”; y con la tabla “*in\_relationNodes*” con el atributo “*idNode*”. El atributo “*nodeFinal*” tiene como valores posibles uno y cero. De esta forma, señalamos si el nodo en cuestión es un punto al que el usuario se puede dirigir en el mapa cuando la aplicación le permite elegir destino. Esto es, hay dos tipos de nodos (finales y no finales), el usuario únicamente puede elegir dirigirse a un destino cuyo campo “*nodeFinal*” tenga el valor 1. Un ejemplo de consulta a esta tabla, para conocer los destinos finales a los que se puede dirigir el usuario es:

*select idNode, Photo from in\_nodes where idMap=2 and nodefinal=1*

Campo	Tipo	Longitud
<b>idNode</b>	mediumint	10
<b>idMap</b>	mediumint	10
<b>idUser</b>	mediumint	10
<b>nodeName</b>	varchar	20
<b>coordenadaX</b>	float	
<b>coordenadaY</b>	float	
<b>nodeFinal</b>	bit	1
<b>Photo</b>	varchar	100

**Tabla 1: Tabla de nodos de la base de datos**

2. Tabla “*in\_maps*”: Esta tabla identifica de manera simple un mapa. Un mapa se compone de N nodos conectados entre sí, que el usuario de la aplicación irá visitando hasta llegar a su destino.

Campo	Tipo	Longitud
<b>idMap</b>	mediumint	10
<b>mapName</b>	varchar	20
<b>Photo</b>	varchar	100

**Tabla 2: Tabla de mapas de la base de datos**

3. Tabla “*in\_relationNodes*”: En esta tabla se guardan las relaciones que tienen dos nodos de un mapa entre sí. La clave primaria de la tabla la forman los atributos “*idNodeInitial*”, “*idNodeFinal*” y “*idMap*”. De esta forma, no existirán en un mismo mapa más de una relación entre dos nodos, teniendo en cuenta los nodos inicial y final de la relación. El atributo “*photo*” de esta tabla guarda la imagen que se mostrará al usuario cuando en el dispositivo se le informe del siguiente punto de la ruta que tiene que visitar. Para ayudar al

usuario, se mostrarán los nodos desde diferentes vistas. Es decir, si un usuario debe visitar el nodo tres y viene desde el nodo cuatro, la imagen mostrada en su *smartphone* será del punto tres vista desde donde se encuentra el usuario, es decir, desde el nodo cuatro del mapa. El campo “*idUser*” identifica al usuario que tiene esta relación establecida entre dos nodos para seguir en la ruta.

Se ha pensado en la posibilidad de incorporar rutas específicas para determinados usuarios en función de la experiencia de sus tutores y educadores. Se tiene en cuenta este aspecto en la ruta propuesta al usuario, si el tutor de un usuario considera que, por ejemplo, para ir del laboratorio 12 a la cafetería debe, por diferentes motivos, pasar por un determinado punto que la ruta calculada por la aplicación no incluye. Para gestionar esto, en esta tabla de la base de datos, el atributo “*idUser*” incorpora un usuario genérico identificado con id igual a cero, que será el usuario que no tenga ninguna relación a tener en cuenta en el cálculo de la ruta entre dos nodos. Esta decisión se ha tomado con el objetivo de facilitar la ruta del usuario hasta su destino. Un ejemplo de consulta para saber el siguiente nodo al que se tiene que dirigir el usuario:

*select idNodeFinal from in\_relationNodes where idNodeInitial=5 and  
idMap=2 and idUser=0*

Campo	Tipo	Longitud
<b>idUser</b>	mediumint	10
<b>idNodeInitial</b>	mediumint	10
<b>idNodeFinal</b>	mediumint	10
<b>idMap</b>	mediumint	10
<b>Photo</b>	varchar	40

**Tabla 3: Tabla de aristas entre nodos**





# 7 Pruebas

---

Las pruebas realizadas en el proyecto pueden dividirse en dos grandes apartados:

1. Cliente
2. Servidor

## 7.1 Pruebas cliente

En el cliente del sistema, es decir la aplicación Android, se han realizado pruebas para comprobar su correcto funcionamiento a medida que se iban desarrollando los diferentes módulos y las pantallas que los componen.

Se han incluido pruebas de caja negra, donde se prueba si la salida esperada en diferentes métodos se corresponde con la salida real. Además de estas, las pruebas de caja blanca nos han permitido probar módulos y métodos en detalle, con el objetivo de dar solución a todos los posibles problemas.

Con los errores solucionados, se proceden a realizar pruebas generales de la aplicación, donde se prueba el uso completo de la aplicación.

Dado que se trata de una aplicación que va a guiar a personas por entornos interiores, probar diferentes rutas es muy importante. Estas están representadas mediante grafos conexos donde cada arista que une dos nodos se entiende como un posible camino para que el usuario llegue desde un punto al siguiente.

Como se ha explicado en el capítulo de implementación, Paso a paso entre actividades, existen conexiones entre dos puntos que serán específicas de un usuario. Es decir, en un mismo grafo de un mapa, para llegar desde un punto a otro, dos usuarios pueden seguir diferentes rutas si su tutor así lo ha considerado. En este sentido, se han realizado pruebas para un mismo grafo con diferentes usuarios donde seguían rutas diferentes para llegar a un mismo destino.

A continuación se muestra en forma de lista las pruebas realizadas de la aplicación, con el resultado obtenido de cada prueba.

1. **Instalación de la aplicación:** El sistema requiere de conexión a internet la primera vez y avisa al usuario.
2. **Inicio de la aplicación sin conexión a internet:** Una vez instalada, el sistema arranca correctamente.

3. **Inicio de la aplicación con conexión a internet:** El sistema conecta con el servidor y se descarga la última versión de la base de datos.
4. **Usuarios de un tutor:** Dado un “*idTutor*” el sistema únicamente carga los usuarios que están a su cargo.
5. **Carga de información de usuarios:** El sistema carga de forma correcta la información, foto y nombre identificativo, de cada usuario de la base de datos.
6. **Usuarios seleccionables:** Todos los usuarios cargados son seleccionables. Se puede navegar entre ellos hasta realizar una selección.
7. **Usuario reconocido:** Una vez se ha elegido, el sistema reconoce al usuario hasta el final de la aplicación.
8. **Instalación de aplicación lector QR:** El sistema comprueba si el dispositivo dispone de la aplicación BarCode Scanner instalada. En caso negativo, se informa al usuario que debe instalarla.
9. **Uso de la aplicación BarCodeScanner:** Después de la instalación, el sistema no vuelve a informar al usuario. Se abre la aplicación para escanear códigos QR.
10. **Destinos disponibles:** El sistema carga correctamente los destinos que puede elegir el usuario. Se comprueba que los destinos cuyo atributo “*nodeFinal*” sea cero en la tabla “*in\_nodes*” de la base de datos, no aparecen como posibles destinos finales.
11. **Carga de destinos:** La información de los destinos disponibles, foto y nombre descriptivo, se carga correctamente y se muestra al usuario en forma de lista.
12. **Selección de destino:** Cuando el usuario hace una selección, el sistema reconoce y guarda donde se quiere dirigir el usuario.
13. **Cálculo de la ruta:** Se comprueba que dado un punto de origen, y el destino elegido, el sistema calcula correctamente la ruta para el usuario de la aplicación.
14. **Lista de pistas:** El sistema va ofreciendo en orden las pistas que debe seguir el usuario.
15. **Retroceder un punto:** La aplicación permite al usuario retroceder en caso de haberse perdido hasta un punto anterior de la ruta. El sistema recalcula su ruta hasta el destino desde este punto.

16. **Mensajes de resultado de pistas:** El sistema va informando al usuario según va siguiendo las pistas. Se prueba que en función de si el usuario acierta o falla, el sistema informa correctamente.
17. **Registro de actividad:** Se prueba que cualquier actividad del usuario queda registrada.
18. **Envío de registro de actividad:** Una vez el usuario ha finalizado el uso de la aplicación, el sistema guarda su registro de actividad y se envía por correo electrónico al servidor de AmILab.
19. **TextToSpeech:** Se ha probado que en cada una de las pantallas de la aplicación, se activa el asistente de voz que va guiando al usuario durante el trayecto. Según la pantalla de la aplicación, el asistente informa con una u otra información.
20. **Cierre de la aplicación:** Cuando el usuario ha finalizado su uso, la aplicación se cierra correctamente.

### 7.1.1 Ejemplo de ruta

En este apartado del capítulo de pruebas se van a detallar las realizadas referentes a las diferentes rutas que se han probado del grafo Ilustración 23: Grafo representativo de mapa de AssisT-In.

Se han probado en este mapa dos usuarios:

- **Usuario 1:** Usuario normal con las conexiones representadas por líneas continuas, que representan la unión entre dos puntos del mapa.
- **Usuario 2:** Usuario cuyo tutor ha considerado incluir algunas rutas específicas para ir de un punto a otro. Además, dispone del resto de conexiones de las que tiene un usuario normal.

A continuación se plantea un ejemplo de ruta para ambos usuarios. La ruta que deben seguir los usuarios es del punto uno al punto siete del mapa.

1. **Usuario 1:** Parte desde el nodo uno. Mediante el algoritmo BFS, se conocen sus nodos vecinos (nodo tres). El usuario buscará la pista en el punto tres. Desde aquí, es decir, una vez escaneado este punto, el sistema informará según el resultado del QR escaneado. Hay tres opciones de resultado, que se pueden ver en la Ilustración 13: Imágenes finales de la aplicación, según si el usuario ha escaneado una pista correcta, se ha equivocado de pista en la ruta o ha escaneado una etiqueta incorrecta. Desde el punto actual, se sabe que los posibles puntos que puede visitar el usuario son el dos y el cuatro. Puesto que desde el nodo dos, se pasará por el cuatro posteriormente, el usuario se tendrá

que dirigir directamente a este último. Nuevamente la aplicación reconocerá la etiqueta escaneada e informará sobre el resultado. Una vez aquí, el usuario tiene la opción de ir al punto cinco o al punto siete. Puesto que el destino elegido está entre los destinos inmediatos posibles, se dirigirá directamente hasta este punto donde finaliza su ruta. Aquí el sistema informará al usuario que ha finalizado con éxito, como se ve en la última imagen de la Ilustración 13: Imágenes finales de la aplicación. Para finalizar, la aplicación guarda el registro de actividad del usuario y lo enviará por correo al servidor.

2. **Usuario 2:** Como se observa en el grafo, el tutor de este usuario ha establecido dos conexiones específicas para él. Estas dos uniones enlazan los puntos cuatro y cinco, con el punto inicial. Cuando el sistema calcule la ruta para el usuario, tendrá una prioridad mayor cualquiera de estas dos conexiones. Por tanto la ruta que seguirá es la siguiente. Desde el punto de origen, podrá visitar los puntos tres, cuatro y cinco dónde los dos últimos tienen mayor prioridad. Desde el punto cinco, en caso de visitar, se tendrá que pasar posteriormente por el punto cuatro para llegar al destino, por lo que el usuario tendrá que dirigirse directamente hasta el nodo cuatro. Desde aquí, y dado que ya se habían analizado los puntos tres y cinco en el paso anterior, el usuario puede dirigirse hacia los nodos dos y siete. Dado que el destino elegido es este último, el sistema mostrará por pantalla que se debe visitar este punto, dónde se pone fin a la ruta del usuario.

Igual que en el usuario 1, el sistema irá informando por pantalla y voz del resultado en cada punto visitado. En caso de error de ruta, se volverá a mostrar el siguiente punto que debe ser visitado.

Las diferentes rutas probadas en la aplicación se han efectuado con éxito. En cada una de ellas se tenían en cuenta las especificaciones de los tutores en caso de haberlas, y en su defecto se establecía la ruta calculada por el algoritmo BFS estándar.

Respecto a las rutas, también se ha probado el caso de que el usuario se pierda en un determinado momento y decida volver a un punto anterior. En este caso el sistema solventó el problema, volviendo a informar al usuario de la siguiente pista a buscar desde el punto en el que se encuentra actualmente.

## **7.2 Pruebas servidor**

En todo sistema con una arquitectura cliente-servidor han de probarse ambas partes para comprobar el funcionamiento. En la parte servidor se prueba que la aplicación utiliza correctamente todos los servicios.

Diferentes métodos de la aplicación requieren de servicios proporcionados por el servidor. Durante el desarrollo de la aplicación se ha configurado un servidor local, que

posteriormente será sustituido por el servidor del laboratorio AmILab. En local se han probado todas las funcionalidades que posteriormente se ejecutarán en el servidor de AmILab.

Basándonos en el apartado Paso a paso entre actividades donde se detalla el proceso que sigue el uso de la aplicación, se explica a continuación las pruebas realizadas del servidor.

En primer lugar, la aplicación se inicia comprobando si se dispone de conexión a internet. En caso afirmativo el sistema se comunica con el servidor mediante una petición http para la descarga de la base de datos local alojada en este. Para esto se requiere del procedimiento almacenado “*dump.php*”.

```
<?php
error_reporting(-1);
mb_internal_encoding("UTF-8");
mb_http_output("UTF-8");

$mysqlDatabaseName = 'assist';
$mysqlUserName = 'root';
$mysqlPassword = 'password';
$mysqlHostName = '150.244.57.130';

$comando='./mysql2sqlite.sh -u root -ppassword assist users
in_nodes in_maps in_relationNode';

$ret = system($comando, $output);
?>
```

### **Ilustración 27: Fichero dump.php**

Como se puede observar en la imagen anterior, se configura la conexión con la base de datos y se ejecuta el script “*mysql2sqlite.sh*” que convierte la base de datos al formato admitido por el dispositivo Android. Una prueba rápida es ejecutar una llamada desde el navegador para comprobar que el servidor está funcionando y nos devuelve el volcado de la base de datos local. La llamada sería la siguiente, y el resultado se puede ver en el ANEXO B: Volcado base de datos:

<http://127.0.0.1/AssistIn/dump.php>

Una vez tenemos la base de datos cargada en nuestro dispositivo, la aplicación mostrará en este orden, las pantallas de elección de usuario y elección de destino. Las imágenes que se le muestran al usuario en estas pantallas están alojadas en el servidor. Se procede a la descarga, como se explicó en el capítulo 6.2 del documento, mediante “*getImage.php*”. Para comprobar de forma manual el correcto funcionamiento de este, se procede a realizar una llamada como la siguiente desde el navegador, y se comprueba su salida:

[http://127.0.0.1/AssistIn/user\\_data/photos/antonio.jpg](http://127.0.0.1/AssistIn/user_data/photos/antonio.jpg)

Una vez realizadas las pruebas en el servidor local, se procede a probar la aplicación en el servidor alojado en el laboratorio AmILab. El funcionamiento debe ser igual que en local. Para ello, se configura la aplicación para que realice las llamadas a la dirección IP correspondiente al servidor remoto.

## 8 Conclusiones

---

El objetivo principal de AssisT-In es el guiado de personas con diversidad funcional intelectual por entornos interiores, para promover su independencia.

El sistema desarrollado utiliza una herramienta muy habitual hoy en día, como es un *smartphone*, lo que facilita su utilización por parte del usuario. La aplicación desarrollada se basa en dividir la ruta en una secuencia de pasos, mediante la identificación de marcas visuales. Se ha trabajado en que la aplicación cuente con los elementos imprescindibles para ayudar al usuario.

La interfaz de la aplicación se ha diseñado basándonos en la experiencia de tutores y educadores de personas con diversidad funcional intelectual para que el usuario se encuentre con el menor número de problemas posibles. De esta forma irá ganando en confianza para los usos posteriores.

Para facilitar la adaptación de este tipo de usuarios, se ha añadido un asistente de voz. Este hará, junto con la información mostrada por pantalla, que el usuario sea capaz de comprender con más facilidad las pistas para conseguir su objetivo de llegar al destino elegido.

La aplicación se ha implementado en base a una arquitectura cliente-servidor. El uso del servidor únicamente se limita a que el dispositivo del usuario tenga conexión a internet. En ese caso se descargará la base de datos actualizada y se iniciará la aplicación. En caso de no existir conexión, el sistema utilizará la última base de datos descargada en el *smartphone* del usuario. Esto es una ventaja dado que en los entornos en los que se pretende implantar el sistema, es decir, entornos interiores como universidades o edificios de la administración, existe una posibilidad de perder la conexión, por lo que la aplicación dejaría de funcionar correctamente.

El sistema sigue un diseño iterativo, similar al DCU. En este tipo de proyectos donde el usuario final de la aplicación no puede aportar la información necesaria, es importante identificar una persona responsable de la realimentación. En este caso, estas personas serán los tutores de los usuarios. Gracias a la experiencia se tendrá información suficiente para las posibles mejoras de la aplicación. Para que los tutores dispongan de los datos del usuario durante el uso de la aplicación, el sistema incorpora un registro de actividad en el que se refleja todo lo que hace el usuario con el dispositivo. Este registro se guardará y enviará por correo para un posterior análisis.

## 8.1 Trabajo futuro

Como trabajo futuro se propone:

- **Evaluar con usuarios:** como idea más importante para el futuro se pretende probar la aplicación móvil con usuarios finales. Esto es, desde el laboratorio AmILab donde se ha desarrollado el proyecto se tiene contacto con expertos en las que se trabaja con estos usuarios. Se quiere hacer una evaluación completa del proyecto con estos usuarios, aunque esta aplicación ha sido desarrollada basándonos en la experiencia de los integrantes del laboratorio.
- **Plan de mantenimiento:** como cualquier aplicación móvil, con el paso del tiempo tanto los dispositivos como el sistema operativo que llevan estos evoluciona y se actualiza, y hay que planificar un mantenimiento. En relación con esto la idea es diseñar un plan que ofrezca soporte a la aplicación para ir actualizándose cada cierto tiempo.
- **Comunicación tutor-usuario:** se ha pensado también, en diseñar y establecer una forma para que el tutor esté comunicado con el usuario en tiempo real. De esta manera se podría hacer un seguimiento en directo de la evolución del usuario. Esto ayudaría también a este en caso de pérdida durante un trayecto.
- **Mejorar las pistas indicaciones:** Aunque las instrucciones proporcionadas se basan en la experiencia de haber trabajado con este tipo de usuarios desde AmILab, siempre puede ser una buena idea actualizarse en favor de los usuarios finales de la aplicación. Esto afectaría tanto a las instrucciones visuales, como puedan ser textos o pictogramas mostrados, como al asistente de voz que guía durante la ruta.



## 9 Bibliografía

---

- [1] Organización Mundial de la Salud. (2015, Enero) OMS Organización Mundial de la Salud. [Online]. <http://www.who.int/classifications/icf/en/>
- [2] Javier y Lobato, Manuel Romañach. (2005) Foro de vida independiente. [Online]. <http://www.forovidaindependiente.org/node/45>
- [3] Laboratorio de Inteligencia Ambiental. (2012, Mar.) AmILab - UAM. [Online]. <http://amilab.ii.uam.es>
- [4] Javier Pérez Ávilas, "SISTEMA DE NAVEGACIÓN ADAPTADO A PERSONAS CON DISCAPACIDAD INTELECTUAL SOBRE DISPOSITIVOS MÓVILES ," Universidad Autónoma de Madrid, Proyecto Fin de Carrera 2012.
- [5] Denso Wave Incorporated, "code QR," Software, 2007.
- [6] American Association on Intellectual and Developmental Disabilities. (2013) AAIDD - Resources for Intellectual and Developmental Disability Professionals. [Online]. <http://aaidd.org>
- [7] "Real Decreto Legislativo 1/2013, de 29 de noviembre, por el que se aprueba el Texto Refundido de la Ley General de derechos de las personas con discapacidad y de su inclusión social," Ley 2013.
- [8] AENOR, "ASSISTIVE PRODUCTS FOR PERSONS WITH DISABILITY - CLASSIFICATION AND TERMINOLOGY (ISO 9999:2011)," AENOR, Norma 9999, 2012.
- [9] Fundación AUNA. (2013) [Online]. <http://www.cermi.es/es-ES/Biblioteca/Lists/Publicaciones/Attachments/36/Estudio20DiscapacidadTICFundAuna.pdf>
- [10] David J. Browne et al., "Designing location-based learning experiences for people with intellectual disabilities and additional sensory impairments," *computers & Education*, vol.

56, no. 1, pp. 11-20, Jan. 2011.

- [11] Stefan Carmien, Rogerio DePaula, Andrew Gorman, and Anja Kintsch, "Increasing workplace independence for people with cognitive disabilities by leveraging distributed cognition among caregivers and clients," pp. 95-104, 2003.
- [12] D.Braddock, M. Thompson M.C. Rizzolo, and R. Bell, "Emerging technologies and cognitive disability. journal of Special Education Technology," vol. 19, no. 4, pp. 49-56, 2004.
- [13] Vicomtech. (2013) Visual interaction & Communications Technologies. [Online]. <http://www.vicomtech.org/pr200/oferta-idi-proyecto-argus-sistema-de-guiado-personal-para-peatones-con-discapacidad-visual>
- [14] Departamento de Tecnología Electrónica de la URJC. (2013) Unidad de información científica de la URJC. [Online]. <http://www.ucci.urjc.es/el-proyecto-ver-con-las-manos-presente-en-foros-tecnologicos/>
- [15] Fundación Orange. Fundación Orange - CITI. [Online]. [http://fundacionorange.es/fundacionorange/proyectos/proyecto\\_citi.html](http://fundacionorange.es/fundacionorange/proyectos/proyecto_citi.html)
- [16] Fundación Orange, la Fundación Síndrome de Down de Madrid y la Universidad Carlos III de Madrid. El proyecto BIT - Nuevas tecnologías aplicadas a la educación. [Online]. <http://www.educacontic.es/ca/blog/el-proyecto-bit>
- [17] SHIH-KAI TSAI and Chung Yuan, "WADER: a novel wayfinding system with deviation recovery for individuals with cognitive impairments," pp. 267-268, 2007.
- [18] Daniel K. Davies, Steven E. Stock, Shane Holloway, and Michael L. Wehmeyer, "Evaluating a GPS-Based Transportation Device to Support Independent Bus Travel by People With Intellectual Disability," vol. 48, no. 6, Dec. 2010.
- [19] IEEE Standards Board, "IEEE Standard Glossary of Software Engineering Terminology," Standard 1990.
- [20] The Eclipse Foundation. (2015) Eclipse - The Eclipse Foundation. [Online].

<https://www.eclipse.org>

- [21] Balsamiq Studios. (2008) Balsamiq Mockups - Balsamiq. [Online]. <https://balsamiq.com/products/mockups/>
- [22] Google. (2008, Sep.) Android Software Development Kit. [Online]. <https://developer.android.com/sdk/>
- [23] Genymobile. (2014) Genymotion. [Online]. <https://www.genymotion.com/>
- [24] SequelPro & CocoaMySQL Teams. (2002) Sequel Pro. [Online]. <http://www.sequelpro.com>
- [25] SourceForge.net. (2003) PHPMyAdmin. [Online]. <http://www.phpmyadmin.net>
- [26] The Omni Group. (1994) OmniGraffle - diagramming and graphic design for Mac, iPhone and iPad - The OmniGroup. [Online]. <https://www.omnigroup.com/omnigraffle>
- [27] Norman, *The Psychology Of Everyday Things.*, 1988.
- [28] AENOR, "Human-centred design processes for interactive systems," AENOR, Norma 1999.
- [29] Gobierno de Aragón. (2015) ARASAAC - Portal Aragonés de la Comunicación Aumentativa y Alternativa. [Online]. <http://www.catedu.es/arasaac/>



## Anexo A: Algoritmo Breadth First Search

---

Teniendo la matriz del grafo construida, a continuación se muestra el pseudocódigo del algoritmo BFS utilizado que nos dará la ruta de puntos a recorrer desde un nodo inicial a un nodo final del mapa.

```
recorridoAnchura(nodoInicial, nodoFinal) {  
    Q = crearCola();  
    añadirACola(nodoInicial);  
    Mientras (Q != vacia){  
        añadirARuta(nodoActual);  
        nodoActual = extraerDeCola();  
        si(actual == nodoFinal){  
            añadirARuta(nodoActual);  
            break;  
        }  
        nodosVisitados[nodoActual] = true;  
        for(desde 0; hasta numeroNodos; +1){  
            v = grafo[nodoActual][posicionBucle]  
            si (v==3 y noVisitado y noVisitadoRutaEspecifica){  
                visitadoRutaEspecifica[posicionBucle] = true  
                añadirACola(posicionBucle)  
                recorrido[posicionBucle] = nodoActual  
            }  
        }  
    }  
    for(desde 0; hasta numeroNodos; +1){  
        v = grafo[nodoActual][posicionBucle]  
        si (v==1 y noVisitado y noVisitadoRutaEspecifica){  
            visitadoRutaEspecifica[posicionBucle] = true
```

```

        añadirACola(posicionBucle)

        recorrido[posicionBucle] = nodoActual
    }

}

ruta = imprimirRecorrido(nodoInicial, nodoFinal)

devolver ruta;

}

```

A continuación se muestra la matriz representativa de un grafo correspondiente a un mapa de la aplicación.

	1	2	3	4	5	6	7
1	2		1	3	3		
2		2	1	1			
3	1	1	2	1			
4	3	1	1	2	1		1
5	3			1	2		
6						2	1
7						1	2

**Tabla 4: Matriz de un grafo representativo de un mapa**

# ANEXO B: Volcado base de datos

```
127.0.0.1/Assistin/dump.php
PRAGMA synchronous = OFF; PRAGMA journal_mode = MEMORY; BEGIN TRANSACTION; CREATE TABLE "users" ( "idUser" mediumint(10) NOT NULL , "userName" varchar(20) NOT NULL, "userLastName" varchar(20) NOT NULL, "gender" varchar(10) NOT NULL, "tutorName" varchar(20) NOT NULL, "idTutor" mediumint(10) NOT NULL, "photo" varchar(100) NOT NULL DEFAULT 'default.jpg', PRIMARY KEY ("idUser") ); INSERT INTO "users" VALUES (0,'Pepe','Martin','masc','Marcos','2','josiko1.jpg'); INSERT INTO "users" VALUES (1,'Jose','Moreno','masc','Mario','1','josiko1.jpg'); INSERT INTO "users" VALUES (2,'Mario','Jimenez','masc','Marcos','2','josiko2.jpg'); INSERT INTO "users" VALUES (3,'Raul','Gonzalez','masc','Marcos','2','josiko1.jpg'); CREATE TABLE "in_nodos" ( "idNode" mediumint(10) NOT NULL, "idMap" mediumint(10) NOT NULL, "idUser" mediumint(10) DEFAULT '0', "nodeName" varchar(20) NOT NULL, "coordenadaX" float NOT NULL, "coordenadaY" float NOT NULL, "nodeFinal" tinyint(1) DEFAULT NULL, "photo" varchar(100) NOT NULL DEFAULT 'default.jpg', PRIMARY KEY ("idNode","idMap") CONSTRAINT "map_node" FOREIGN KEY ("idMap") REFERENCES "in_maps" ("idMap") ON DELETE CASCADE, CONSTRAINT "users_node" FOREIGN KEY ("idUser") REFERENCES "users" ("idUser") ON DELETE CASCADE ); INSERT INTO "in_nodos" VALUES (1,2,0,'aula1','0,0,1','aula1.jpg'); INSERT INTO "in_nodos" VALUES (2,2,0,'aula2','0,0,1','aula2.png'); INSERT INTO "in_nodos" VALUES (3,2,0,'aula3','0,0,1','aula3.jpg'); INSERT INTO "in_nodos" VALUES (4,2,0,'aula4','0,0,0','aula4.jpg'); INSERT INTO "in_nodos" VALUES (5,2,0,'aula5','0,0,1','aula5.jpg'); INSERT INTO "in_nodos" VALUES (6,2,0,'aula6','0,0,1','aula6.jpg'); INSERT INTO "in_nodos" VALUES (7,2,0,'aula7','0,0,1','aula7.jpg'); CREATE TABLE "in_maps" ( "idMap" mediumint(10) NOT NULL, "mapName" varchar(20) NOT NULL, "photo" varchar(100) NOT NULL DEFAULT 'default.jpg', PRIMARY KEY ("idMap") ); INSERT INTO "in_maps" VALUES (1,'mapa1',''); INSERT INTO "in_maps" VALUES (2,'mapa2',''); CREATE TABLE "in_relationNodes" ( "idUser" mediumint(10) NOT NULL, "idNodeInitial" mediumint(10) NOT NULL, "idNodeFinal" mediumint(10) NOT NULL, "idMap" mediumint(10) NOT NULL, "photo" varchar(40) DEFAULT NULL, PRIMARY KEY ("idNodeInitial","idNodeFinal","idMap") CONSTRAINT "idMap" FOREIGN KEY ("idMap") REFERENCES "in_maps" ("idMap") ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT "nodoFinal" FOREIGN KEY ("idNodeFinal") REFERENCES "in_nodos" ("idNode") ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT "nodoInicial" FOREIGN KEY ("idNodeInitial") REFERENCES "in_nodos" ("idNode") ON DELETE CASCADE ON UPDATE CASCADE ); INSERT INTO "in_relationNodes" VALUES (0,1,3,2,'aula1_3.jpg'); INSERT INTO "in_relationNodes" VALUES (2,1,4,2,'aula1_4.jpg'); INSERT INTO "in_relationNodes" VALUES (2,1,5,2,'aula1_5.jpg'); INSERT INTO "in_relationNodes" VALUES (0,2,3,2,'aula2_3.png'); INSERT INTO "in_relationNodes" VALUES (0,2,4,2,'aula2_4.png'); INSERT INTO "in_relationNodes" VALUES (0,3,1,2,'aula3_1.jpg'); INSERT INTO "in_relationNodes" VALUES (0,3,2,2,'aula3_2.jpg'); INSERT INTO "in_relationNodes" VALUES (0,3,4,2,'aula3_4.jpg'); INSERT INTO "in_relationNodes" VALUES (2,4,1,2,'aula4_1.jpg'); INSERT INTO "in_relationNodes" VALUES (0,4,2,2,'aula4_2.jpg'); INSERT INTO "in_relationNodes" VALUES (0,4,3,2,'aula4_3.jpg'); INSERT INTO "in_relationNodes" VALUES (0,4,5,2,'aula4_5.jpg'); INSERT INTO "in_relationNodes" VALUES (0,4,7,2,'aula4_7.jpg'); INSERT INTO "in_relationNodes" VALUES (2,5,1,2,'aula5_1.jpg'); INSERT INTO "in_relationNodes" VALUES (0,5,4,2,'aula5_4.jpg'); INSERT INTO "in_relationNodes" VALUES (0,6,7,2,'aula6_7.jpg'); INSERT INTO "in_relationNodes" VALUES (0,7,4,2,'aula7_4.jpg'); INSERT INTO "in_relationNodes" VALUES (0,7,6,2,'aula7_6.jpg'); CREATE INDEX "in_relationNodes_idMap" ON "in_relationNodes" ("idMap"); CREATE INDEX "in_relationNodes_nodoFinal" ON "in_relationNodes" ("idNodeFinal"); CREATE INDEX "in_maps_nodeName" ON "in_maps" ("idMap"); CREATE INDEX "users_userName" ON "users" ("userName"); CREATE INDEX "users_idTutor" ON "users" ("idTutor"); CREATE INDEX "users_gender" ON "users" ("gender"); CREATE INDEX "in_nodos_map_node" ON "in_nodos" ("idMap"); CREATE INDEX "in_nodos_users_node" ON "in_nodos" ("idUser"); END TRANSACTION;
```

Ilustración 28: Volcado de la base de datos